中华学习机

紫金Ⅱ系列

APPLE Ⅱ

Zh

图形管理

中华学习机图形管理

朱国江　杜晓荣
詹丰兴　孙建隆　编著

气象出版社

内 容 简 介

本书系统地介绍了交互式绘图程序设计、实用图形管理与技巧、机器语言绘图技术等内容。选题广泛，取材新颖，叙述详尽。

本书与《中华学习机编程技巧》(气象出版社，1988年2月出版)、《中华学习机数据处理》(气象出版社，1989年12月出版)和《中华学习机汉字软件》(气象出版社，1990年2月出版)等成系列书。本书以国家教育委员会指定推广的优秀机型——中华学习机为选型机，适用于APPLEⅡ兼容机，紫金Ⅱ系列机。

本书可供大专院校师生，工程技术人员，微机应用、管理人员和广大青少年参阅，也可作为各类培训班的教学参考书。

中华学习机图形管理

朱国江　杜晓荣　詹丰兴　孙建隆　编　著

责任编辑　黄丽荣

气象出版社出版

(北京西郊白石桥路46号)

新华书店总店科技发行所发行　全国各地新华书店经销

北京印刷一厂印刷

开本：787×1092 1/32　印张：18

字数：382 千字　印数：1—8000

1991年2月第一版　1991年2月第一次印刷

ISBN 7-5029-0425-5/TP·0022　定价：9.10

前　　言

计算机图形学以其强大的生命力，正在许多领域得到日益广泛的普及和应用。机械制造业的ＣＡＤ系统、电气工程的布局设计、建筑学上的平面布置、市政建设规划安排，正在越来越多地使用图形学这一有力工具。在自然科学方面，生物学、物理学、化学、测绘学、气象学等学科，利用计算机图形显示将使问题变得更加直观和容易理解。艺术家们运用计算机图形学这个现代化方法，进行图案设计、动画制作，将使作品更加丰富多采，栩栩如生。在商业和行政管理上，运用曲线图、直方图等可使枯燥烦杂的统计数据变得一目了然。对决策机构，比起令人目眩的数字来更会受到欢迎。心理学的专家研究表明，具有动画、声音的冒险，战略对抗，战斗模拟和博弈游戏对提高儿童智商有着重要的作用。而在计算机辅助教学ＣＡＩ中，先进的图文并茂、声图并茂技术，更有广阔的应用前景。

本书提供的图形管理方面的程序设计、方法思路、实用技巧和工具软件，旨在加深对图形学基本内容的了解，掌握运用图形语言和有关算法，以及生成各种画面的基本技巧，为进一步开展ＣＡＤ研究和计算机图形学的深入应用打下基础。从这个角度来说，本书是关于描述计算机图形学的深入浅出的实用参考书。

本书以大量的实例，多种选题，介绍BASIC语言和6502机器语言在图形设计和图形管理方面的应用。取材新颖，

文字流畅，叙述详尽，循序渐近。书中提供了一定数量的工具软件和大量的管理程序，也有作者多年实践经验的总结，其中大部分内容，是现行同类国内外书籍和资料中所没有的，具有较高的应用价值。相信会给广大读者提供切实而有益的帮助。

本书资料主要来自作者编程和教学实践，同时还吸收了国内外的先进经验，也选用了少量书籍和杂志上发表的优秀文章。书中阐述的原理、方法、思路和技巧，对计算机辅助教学、图形处理辅助设计和制造、游戏软件开发、办公室自动化等领域有参考价值。而全部程序和工具软件均经严格调试通过，可以直接付诸使用和移植。

本书共分三篇25章，第一篇全部用BASIC语言编程，由杜晓荣编写，第二篇用BASIC和6502机器语言编程，由朱国江（1,2,3,4,5,9,10章）、孙建隆（6,7,8,11,12章）统一设计，共同调试，分别编写，第三篇用6502机器语言编程，由詹丰兴编写。全书由朱国江统一审校。

系列书的出版，得到中国计算机用户协会紫金分会、武汉天问电脑公司、中国中华学习机普及协会、国家气象局气象出版社、软件报社、南京大学大气科学系邹进上教授、李宗恺教授、周朝辅高级工程师，南京大学出版社花国良同志的热情鼓励和大力支持，在此一并表示谢意。

谨以此书，献给广大青少年计算机教育工作者、微机爱好者，献给关心“中华学习机”事业的广大读者，献给支持中华学习机发展的各界同仁，献给“计算机要从娃娃抓起”光辉题词五周年。

朱国江　1989年5月4日于南京大学

目　　录

第一篇　交互式绘图程序设计……………………………（1）

第一章　绘图原理……………………………………………（1）

　第一节　中华学习机的绘图特性……………………………（1）

　第二节　文本方式下的图形绘制……………………………（4）

　第三节　低解析度图形………………………………………（7）

　第四节　高解析度图形………………………………………（11）

　第五节　交互绘图基本概念…………………………………（16）

第二章　点线法交互程序……………………………………（19）

　第一节　友好的人机界面……………………………………（19）

　第二节　直线的交互实现程序………………………………（21）

　第三节　圆的交互程序………………………………………（24）

　第四节　圆弧的交互实现程序………………………………（26）

　第五节　图形的二维变换交互程序…………………………（30）

　第六节　图形存贮……………………………………………（40）

第三章　造型法生成工具……………………………………（45）

　第一节　手控造型法的交互生成程序………………………（45）

　第二节　按参数来生成造型表的交互程序…………………（49）

　第三节　键盘字符交互生成造型的程序……………………（54）

　第四节　造型表的合并、造型的拼积程序…………………（62）

第四章　图形的组织…………………………………………（68）

　第一节　用数据结构描述图形………………………………（68）

第二节　数据的存贮方式……………………………………（79）
第三节　图形数据的交互生成程序……………………………（82）
第四节　几个应用实例…………………………………………（92）
第五章　动态图形的组织………………………………………（107）
第一节　动态图形的显示原理…………………………………（107）
第二节　动态图形的设计技巧…………………………………（114）
第三节　从数据结构来生成一幅动画…………………………（154）
第六章　通用造型软件包 2.0 版………………………………（160）
第一节　软件结构与功能………………………………………（160）
第二节　软件实现………………………………………………（163）
第三节　通用造型软件包 2.0 版使用说明……………………（167）
第二篇　实用图形管理与技巧…………………………………（184）
第一章　初步知识………………………………………………（184）
第一节　机器语言子程序的存贮和调用………………………（184）
第二节　图形搬家的方法………………………………………（196）
第三节　高分辨率图形的绘制…………………………………（202）
第四节　图形数据的拟合过程…………………………………（210）
第二章　图形变换………………………………………………（214）
第一节　图形的移动……………………………………………（214）
第二节　图形的放缩……………………………………………（222）
第三节　镜象复制………………………………………………（231）
第三章　分页显示………………………………………………（239）
第一节　图形的存贮和调用……………………………………（239）
第二节　图形搬家………………………………………………（241）
第三节　屏幕软开关……………………………………………（242）
第四节　分页和连续显示的原理………………………………（242）

第五节　单幅画面的显示…………………………………（243）
第六节　两幅画面的分页显示…………………………（245）
第七节　三幅画面的分页显示…………………………（246）
第八节　三幅画面分页显示的自动控制………………（247）
第九节　四幅画面的分页显示…………………………（249）
第十节　八幅画面的分页显示…………………………（253）
第十一节　翻页显示………………………………………（256）
第四章　图象处理…………………………………………（264）
第一节　左右易位…………………………………………（264）
第二节　水平镜象…………………………………………（267）
第三节　垂直压缩…………………………………………（269）
第四节　成倍翻番…………………………………………（270）
第五章　合并显示…………………………………………（272）
第一节　两幅画面的合并…………………………………（272）
第二节　合并分页显示……………………………………（276）
第三节　机器语言合并图形的方法………………………（279）
第四节　用BASIC程序合并图形…………………………（280）
第五节　多幅图形的合成…………………………………（283）
第六节　图形合成的简便方法……………………………（291）
第六章　移动显示…………………………………………（295）
第一节　画面结构与其记忆位置的对应关系……………（295）
第二节　由下向上移动显示………………………………（302）
第三节　由上向下移动显示………………………………（310）
第四节　向左、向右移动显示……………………………（316）
第七章　移动清屏…………………………………………（322）
第一节　向上、向下移动清屏……………………………（322）

第二节　向左、向右移动清屏…………………………（328）
第三节　画线清屏技巧………………………………（332）
第八章　趣味显示…………………………………（344）
第一节　拉幕显示与清屏……………………………（345）
第二节　合幕显示与清屏……………………………（350）
第三节　上下拼合显示与清屏………………………（354）
第九章　窗口剪辑…………………………………（362）
第一节　窗口技术……………………………………（362）
第二节　折线剪辑……………………………………（363）
第三节　图形剪辑……………………………………（370）
第四节　图形编辑……………………………………（373）
第十章　造型动画…………………………………（379）
第一节　简单动画显示技术…………………………（379）
第二节　汇编语言调用造型表………………………（385）
第三节　造型图形的动画显示………………………（390）
第四节　多个造型图形的动画显示…………………（397）
第五节　动画显示的键盘操作控制…………………（402）
第六节　造型动画设计实例…………………………（411）
第十一章　多页联动………………………………（418）
第一节　两页图象上下联动显示……………………（418）
第二节　三页图象上下联动显示……………………（422）
第三节　多页图象左移联动显示……………………（436）
第十二章　特殊技巧………………………………（442）
第一节　16K RAM卡简介…………………………（442）
第二节　16K RAM卡使用…………………………（444）
第三节　五页图象绕卷显示…………………………（446）

第四节　压缩存贮简介……………………………………………………(456)
第五节　压缩存贮实例……………………………………………………(459)
第三篇　机器语言绘图技术………………………………………………(463)
第一章　机器语言绘图……………………………………………………(464)
第一节　APPLEII的图形空间………………………………………………(464)
第二节　绘图功能的调用…………………………………………………(465)
第二章　造型的设计和使用………………………………………………(473)
第一节　造型原理及造型表结构…………………………………………(473)
第二节　造型表的构成……………………………………………………(475)
第三节　造型表的使用……………………………………………………(475)
第三章　造型程序设计……………………………………………………(477)
第一节　参数设置…………………………………………………………(477)
第二节　造型表的调入和存贮……………………………………………(479)
第三节　造型表初置………………………………………………………(482)
第四节　建立造型…………………………………………………………(484)
第五节　造型删除…………………………………………………………(505)
第六节　造型内插…………………………………………………………(510)
第七节　图形设计…………………………………………………………(514)
第八节　资料表……………………………………………………………(523)
第九节　操作说明…………………………………………………………(524)
第四章　动画设计及图形的合并、剪辑…………………………………(526)
第一节　简单动画设计……………………………………………………(526)
第二节　页面轮换…………………………………………………………(527)
第三节　造型动画…………………………………………………………(531)
第四节　图形的合并………………………………………………………(533)
第五节　图形的剪辑………………………………………………………(536)

第六节　图形的求反……………………………………（542）
第五章　绘图程序的自动生成软件……………………（544）
第一节　通用程序段和有关指针的设置………………（546）
第二节　光点控制…………………………………………（548）
第三节　几个子程序………………………………………（553）
第六章　绘图与声音………………………………………（556）
第一节　发声的基本原理…………………………………（556）
第二节　机器语言发音子程序……………………………（557）
第三节　绘图与声音………………………………………（558）
第七章　绘图程序的必备功能……………………………（562）
第一节　程序结构清晰……………………………………（562）
第二节　程序设计灵活……………………………………（563）
第三节　图象配有说明……………………………………（565）
第四节　提示注意事项……………………………………（565）
结束语……………………………………………………（566）

第一篇

交互式绘图程序设计

第一章 绘图原理

中华学习机具有在显示器（或电视机）屏幕上显示图形功能。本章主要介绍微型机的绘图特性、BASIC语言绘图原理、造型原理、几种生成图形的方法及交互绘图的基本概念。

第一节 中华学习机的绘图特性

计算机绘图对不同的人来说意义不一样，有些人想到了游戏机，另一些人想到了训练用的模拟器，服装师可以把计算机绘图功能作为一种规划的方法，并在设计服装时用来做图形显示，工程师可利用计算机绘图来分析电子电路、机械结构或辅助设计，而教师可用来作为辅助教学的工具，等等。总之，计算机绘图的应用是广泛而多方面的。

中华学习机与APPLEⅡ微型机兼容，有关绘图语言有浮点BASIC语言、LOGO 语言， SUPER PILOT

语言。图形软件有通用造型软件包，APPLEⅡ绘图系统等等。本篇及所属各章主要介绍以浮点BASIC语言为基础的绘图模式。

一、屏幕显示和屏幕方式

1．屏幕显示

中华学习机可以采用专用显示器,或家庭用黑白电视机、彩色电视机，它既可以显示西文字母、数字、符号和汉字文本，也可以显示图形。

2．屏幕方式

由计算机输入到显示器中的信息在屏幕上的显示方式有五种,即文本方式、低解析度图形方式、高解析度图形方式、图形和文本混合方式及中文方式。所谓解析度是指一个图形表达细微部分能力大小。一个图形可由一个个点组合而成，如果这些点很细密，则图形会非常细致，每一个细节都显示得清楚，这就是说解析度高。如果同一幅图形用很大的点去描绘，使很多细节都无法表示，则解析度低。

二、屏幕存贮方式

计算机的屏幕显示是将存贮在RAM或ROM中的部分信息显示出来。由于这个特性，作为绘图的起点，让我们先讨论存贮器的分配情况。

中华学习机有48KBRAM，16KBROM，共64KB(65536个字节)的内存单元。文本方式和低解析度图形方式放在一个存贮容量1024个单元的内存区中，用以存放一个文本或低解析度屏幕的全部信息。在高解析度图形方式中，由于要显示较多的信息，使用了一个存贮容量为8192单元的另一个内存区。这两个内存区通常称为“页”(PAGE)，后者

是专用来存放图形，故又称为“图形缓冲区”。

显然，如果只用一页内存区来存放信息，不可能达到用以连续显示的要求，因此，为了提高实时显示速度，实际上，每种显示方式都设置了两个容量相同的内存区，分别称为“第一页”和“第二页”。一页用来存放信息，另一页用来取信息显示。两页间循环交替工作，以达到连续显示的目的。各种显示方式在存贮器中的分配情况见表1.1。

表 1.1 各种显示方式内存单元分配表

显示格式	页	首地址		末地址	
		十六进制	十进制	十六进制	十进制
文本低解析度图形	1	$ 400	1204	$7FF	2047
	2	$ 800	2048	$BFF	3071
高解析度图形	1	$2000	8192	$3FFF	16383
	2	$4000	16384	$5FFF	24575

三、屏幕方式的切换

加电后屏幕处于西文 BASIC 控制之下，也就是处于文本方式，为使用方便，在BASIC语言中设置一些命令，可实现屏幕显示方式的转换，这样的命令有：

TEXT　置文本方式

GR　置文本和低解析度图形混合方式

HGR　置文本和高解析度图形混合方式

HGR 2　置高解析度图形方式

上述命令可立即执行，也可前面带行号编在程序中使用。除了上述命令可实行屏幕的切换，中华学习机也可通过软开关来访问内存，也能达到类似的作用。所谓软开关，它有开

和关两种状态，由计算机软件控制，故称之为软开关。程序访问一个指定的内存单元就能进入相应的开关状态，且与访问时对这个内存所写入或读出的数据无关，而且只与它的地址有关。

关于用软开关来实现屏幕的切换，在后绪章节介绍。

第二节　文本方式下的图形绘制

文本绘图是指在TEXT方式下，利用BASIC语句和函数，在某些指定位置上打印字符或符号来绘制图形的。

处于文本方式下，屏幕从上至下为24行，每行可显示40个字符，字符在机内存放采用 ASCII码，可正常显示64个字符。

字符在屏幕上显示有三种方式：正常显示（黑底白字），反显示方式（白底黑字）和闪烁显示方式，可分别通过三个命令NORMAL、INVERSE和FLASH来实现。

这三个命令可立即执行也可前边带行号插在程序之中。

一、光标位置的控制

这里介绍一组语句与PRINT语句相结合，可灵活设计屏幕的文本格式显示。

HOME命令

这个命令的功能为清除整个屏幕，将光标移到文本窗口的左上角。

VTAB语句

格式：VTAB定位值

该语句的功能为，用于屏幕光标垂直方向的定位。定位值由算术表达式确定，其值为1—24，有小数部分自动舍去。

HTAB语句

格式：HTAB定位值

该语句的功能为：用于屏幕光标水平方向的定位。定位值由算术表达式来确定，其值为0—255,41相当0号位置。

TAB函数

格式：TAB（参数值）

该函数功能类似于HTAB语句，参数值由算术表达式确定，其值由0—255，有小数部分舍去，若为零，则按256处理。与HTAB不同的是TAB函数必须放在PRINT语句中，且当算术表达式值小于光标当前所在列号时，TAB函数不起作用，即它不能使光标后退。

POS函数

格式：POS（0）

这个函数给出光标当前位置为当前行的第几个字符位置，左端从零算起。

SPC函数

格式：SPC（8）

这个函数必须用在PRINT语句中，它的功能是在输出项之前插入若干个空格（本例是8），它一边移动，一边擦出扫描过位置上的内容，其空格数由参数式子来确定。

二、文本方式下的图形绘制

由上面介绍的语句和函数，可实现不同字符组成的屏幕格式或不同图形，下面介绍几个实例。

1. 用星号打印一个等腰三角形（见程序GP1101）

```
5  REM    GP1101
10  PRINT  TAB( 20);"*"
```

```
20   FOR I = 1 TO 8
30   PRINT  TAB( 20 - I);"*"; SPC(
      2 * I - 1);"*"
40   NEXT
50   PRINT  SPC( 10);
60   FOR I + 1 TO 19
70   PRINT "*";
80   NEXT
90   END
```

```
]RUN
         *
        * *
       *   *
      *     *
     *       *
    *         *
   *           *
  *             *
 *               *
*******************
```

2. 程序G P1102可以打印正弦曲线

```
]LOAD GP1102
]LIST

5   REM   GP1102
10   DIM D(40,20)
20   FOR X = 1 TO 30
30 Y =  INT ( - 7 *  SIN (X * 360
     / 30))
40 D(X,7 - Y) = 1
50   NEXT X
60   FOR Y = 1 TO 14
70   FOR X = 1 TO 30
```

```
80  IF D(X,Y) = 1 THEN  PRINT  TAB(
      X);"*";
90  NEXT X
100  PRINT
110  NEXT Y
120  END
]RUN
 **         **           *
   *           *        * *

*          *           *
    *           *          *


          *           *
     *           *          *


         *        *  *        *
      *
        **         **          *
```

第三节　低解析度图形

低解析度，是指图形的解析度较低，换句话说，这类图形是由较大的色块来组成，不能画出较细密的图形。

一、低解析度图形屏幕

低解析度屏幕相当于一个直角坐标系，它的原点在屏幕的左上角，X轴沿着顶端一行向右，取值为0—39，Y轴为沿着顶端左边一列向下，取值为0—47。

屏幕之所以被划分为40列、48行，是由于低解析度图形在内存的存贮区借用了文本方式下的屏幕存贮区造成的，这一区域，是内存的1024号到2047号单元。

前节介绍过，文本方式下，屏幕被划分成40×24个字符位置，可见一个字符位置恰好是低解析度图形的上下两个色块。我们知道，字符在内存中是以ASCII码存放的，一个ASCII码占用一个字节，即八位二进制数。实际上，在低解析度图形方式下，图形缓冲区的任一个字节八位被分成的右四位、左四位分别表示上下两个图块。

四位二进制数可表示十进制的0到15这16个数，中华学习机利用这16种码来代表16种不同的颜色，见表1.2。

表 1.2 低解析度图形的颜色

颜色	黑	红	深蓝	紫	深绿	灰1	中蓝	浅蓝	棕	橙	灰2	粉红	浅绿	黄	绿蓝	白
数字	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

二、低解析度图形方式

低解析度有图形方式和图形与文本混合方式两种。

1．混合方式

格式：GR

这个命令一是进入低解析度图形屏幕，屏幕下边留有4行文本区，为画图作好准备；二是清除屏幕，将屏幕置成黑色。

2．图形方式

在进入低解度混合方式，您若想去掉4行文本，使屏幕成为纯图形方式，可用下述命令

POKE －16302， 0 ↙

3．从纯图形方式返回到混合方式

通过访问地址－16301或用“GR”命令，可返回到混

合显示方式。

例 1， POKE －16301， 0

该命令保持当前40列图形不变而将 4 行文字行恢复。

GR

该命令清除屏幕图形后并恢复到混合显示方式。

三、低解析度绘图语句

下面介绍在低解析度图形方式下的绘图语句，其中有置图形颜色、画点、画线及返回坐标点颜色的语句和函数。

1．设定颜色语句

格式：COLOR＝n

为设定颜色的语句。其等号右边的n可为表达式、其值在 0—255之间，超过16的，取被16除后的余数，有小数部分自动舍去。n值的含义见表1.2。

2．在指定位置上画点

格式：PLOT X，Y

为画点语句，表示在（X，Y）坐标上画出一点。该点为一个小的长方形色块。X取值为 0—39，Y取值为 0—47。

3．绘直线语句

① 绘水平线

格式：HLIN X1， X2 AT Y

为画水平线语句，表示从坐标点（X1，Y）到坐标点（X2，Y）画一条水平线。X1、X 2 的取值范围在 0—39，Y取值范围在 0—47。

② 画垂直线

格式：VLIN Y1， Y2 AT X

为画垂直线语句，表示从坐标点（X，Y1）到坐标点

（X，Y2）之间画一条垂直线。Y1和Y2的取值范围在0—47之间，X的取值范围为0—39，Y1、Y2和X超出上述范围，计算机将给出出错信息。

4．返回指定坐标点的颜色值

格式：SCRN（X，Y）

这是一个函数，该函数返回低解析度图形方式下（X，Y）坐标点的颜色值。

例如：X＝SCRN（12，30）

将第13列第31行处的点的颜色的代码值放入变量X中。

以上语句，可以写入BASIC程序中，也可以处于立即执行方式，其中X、Y能以算术表达式出现。

四、实例

1．用画点语句画一个等腰三角形（见程序GP1103）

```
]LOAD GP1103
]LIST

5   REM   GP1103
20   GR
30   COLOR=  RND (16) * 16 + 1
40   FOR Y = 20 TO 30
45  X = Y - 10
50   PLOT Y,Y
52   PLOT X,50 - Y
55   PLOT X,30
60   PLOT Y,30
70   NEXT
100   GOTO 30
```

执行程序GP1103，可以在屏幕的中间位置画一个颜色不断改变的等腰三角形。20句为进入低解度图形方式，30句为改变颜色，40—70句循环为用点画出等腰三角形。

2．用画线语句画等腰三角形（见程序GP 1104）

```
5  REM   GP1104
10  GR
20  COLOR= 3
30  FOR J = 20 TO 30
40 X1 = J - 10:X = 30 - J
50  HLIN X1,X AT J
60  VLIN X1,X AT J
70  NEXT
```

执行程序 GP 1104；可以在屏幕的适当位置画两个等腰三角形，这两个三角形都充填紫色。

第四节　高解析度图形

高解析度图形克服了低解析度图形方式的缺点，能绘出较精细的图形，主要是增加屏幕缓冲区的内存来实现。

一、高解析度屏幕

高解析度屏幕坐标原点设在左上角，X轴方向向右，取值范围为 0 到279，Y 轴方向向下取值范围在 0 至191 之间。整幅屏幕由192×280=53760点阵的象素构成。

在低解析度图形方式下，一个字节可显示两个色块，而在高解析度图形方式下，八位二进制的后七位的状态是 1 还是 0 分别表示在 7 个位置上有点还是无点。这样从左到右由40字节可表示屏幕的一行信息，即 280 个象素，由于整幅有192行，故需要40×192 =7680个字节才能完整地表示高解度图形。中华学习机设置了两个图形缓冲区，分别称为第一页和第二页，从8192到 16384 和16384 到24576，每个区域的长

度为8192字节，稍大于7680，这主要是计算地址方便，因为8192的十六进制为＄2000。

一个字节的后七位表示7个象素点，只剩下一位用来控制颜色，那就只能控制两种状态(1或0)，这虽然是不好的。为此对颜色的控制采取了以下补救方法：

高解析度颜色有7种，用语句来设定，对应关系见表1.3。

表 1.3 高解析度图形的颜色

n 值	0	1	2	3	4	5	6	7
颜 色	黑	绿	蓝	白1	黑2	红	黄	白3

需要注意的是，点的颜色由下列因素决定，当点的相应位为0时，对应屏幕为一个黑点，点相应为1时，还由点在屏幕的相对位置来确定显示的颜色：

处在偶数列上的点可能为黑、紫或蓝色。

处在奇数列上的点可能是黑、绿或橙色。

两个紧靠着的颜色点全显示白色。

二、高解析度图形方式

高解析度有图形方式和图形与文本混合方式两种。

1. 混合方式第一页

格式：HGR

该语句置屏幕为高解析度图形与文本混合方式、显示第一页，将屏幕上除最下边的4行作为文本输出外，其余部分设定为黑色。

如果你想显示高解析度第一页混合方式而不清除图形，

可用下述命令

POKE　－16304，0　　选图形方式

POKE　－16297，0　　选高解析度图形

POKE　－16301，0　　选混合方式

2．纯高解析度图形方式

用HGR语句只能进入高解析度与文本显示的混合方式。这种方式的屏幕区显示280×160个点，并在屏幕下面有4行文本显示区。

进入纯图形方式可用下述命令：

第一页

HGR：POKE　－16302，0

第二页

HGR 2

如果你想返回到高解析度图形与文本的混合方式，可执行下述命令：

POKE　－16301，0

三、高解析度绘图语句

1．高解析度的颜色

格式：HCOLOR＝n

该语句置高解析度图形的颜色。在高解度图形方式下有8种颜色，见表1.3，n的取值范围在0—7之间。

例如　HCOLOR＝2

设置高解度图形的颜色为蓝色。

当n值为1、2、5、6的4种颜色，对于不同彩色显示器，其颜色也有所不同。

2．绘图语句

格式：HPLOT　X，Y

为画点的语句。表示在坐标（X，Y）处以最近设定的颜色画出一个点。

格式2：HPLOT　TO X，Y

从上一次画的最后一个点（X，Y）坐标之间画一条直线。

格式3：HPLOT　X1，Y1 TO X2，Y2……TO Xn，Yn

为画折线语句。从（X1，Y1）到（X2，Y2）之间画一条线，再从（X2, Y2）到（X3, Y3）画一条线，依次画下去，直到Xn, Yn 的折线。如果各点离得足够近，就能画近似的曲线，X、Y必须成对出现，其取值范围不能超过屏幕，且X，Y可取算术表达式。

四、实例

1. 用画点的办法，作出一条SIN（X）曲线

程序如GP1105所示。

```
5  REM   GP1105
10  HGR : POKE  - 16302,0
20  HCOLOR= 3
25  HPLOT 0,80 TO 270,80
28  HPLOT 0,0 TO 0,180
30 PI = 3.14159 / 180
40  FOR I = 0 TO 120
50 X = 2 * I:Y = 80 - 60 *  SIN (
     3 * I * PI)
60  HPLOT X,Y
70  NEXT I
80  END
```

程序GP1105所画的图形见图 1.1

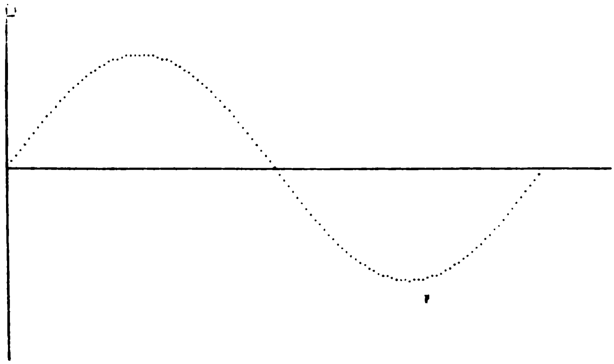


图 1.1 正弦曲线

2. 程序GP 1106，可在屏幕上画出5个同心椭圆（见图1.2）

```
5   REM    GP1106
10 R1 = 15:R2 = 10
20  HGR2 : HCOLOR= 3
30 PI = 3.14159 / 180:X0 = 120:Y0
       = 90
40  FOR I = 1 TO 5
50  FOR J = 1 TO 360 STEP 5
70 X = X0 + R1 * I *  SIN (PI * J
     ):Y = Y0 + R2 * I *  COS (PI
     * J)
80  HPLOT X,Y
90  NEXT J
100  NEXT I
110  END
```

当R1与R2取同值，则可画出5个同心圆，见图1.3 。

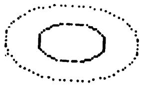


图 1.2　同心椭圆

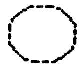


图 1.3　同心圆

第五节　交互绘图基本概念

前面三节介绍了中华学习机几种图形的表示法。低解析度图形分辨低，但显示内存开销小，具有较多颜色；高解析度图形分辨率相对较高，能画较精细的图形，且还具有造型表功能，具有较大灵活性。以后章节主要讨论在高解析度状态下图形显示及图形管理。

事实上，在计算机辅助设计（C A D），计算机辅助教学（C A I）和构造模型中，图形已作为观察和交互手段，正如一句中国颜语“一幅图顶得上一千个字”一样，因为我们可以快速而容易地从说明图或动画过程中抓住一个概念或情况的实质。

在高解析度状态下，中华学习机提供了两种生成图形的方法，点线法和造型法。点线法是用 HPLOT 语句编程生成图形，而造型法是事先按图形轨迹要求制作一张造型表，然后再用 DRAW 语句编程调用。前者可实现函数参数绘图，显示速度慢，后者生成造型表较困难，但显示速度快，可满

足动画的制作。这两种生成图形方法都要编程或者借助于一些辅助性工具制作造型表，这种方式称为“被动式作图”。用这种方式作图，创作一幅图形较麻烦，由于中华学习机内存较小，有时实现一幅复杂图形需要用程序覆盖技术来实现。这样我们想能否不需要编程，让用户通过键盘与屏幕交互作用来满足以下一些要求：

·不需要编程，交互作用生成一幅由 HPLOT 语句绘制的图形，可对这幅图形移动，放大处理，且能保存这幅图形。

·不需要编程，交互作用生成一幅由 HPLOT 语句绘制的图形，同时也能将显示的图形转换成相应的造型表格式来存贮。

·不需要编程，用交互作用的方式对造型表中显示的子图形实现动态地插入，复制一幅图形或生成指定某个造型在屏幕上移动的动画轨迹。

回答是肯定的，我们在中华学习机根据友好的人机界面要求，按照模块化程序设计思想，编制了一个实用的交互图形工具——通用造型软件包，图1.4为在中华学习机上借用该工具软件，完成的减速器拆装模拟系统过程的某个动画时刻的硬拷贝，从图形的制作到动画生成，完全交互完成，在拆卸或装配过程中，所有的信息都装入内存，不与软盘交换信息。

本篇的第二章主要围绕这个造型工具，介绍交互绘图程序的建立以及编程思想，并给出一些实用的交互绘图，交互生成造型的程序。

所谓交互式绘图，即用户可以借助于交互设备，例如键

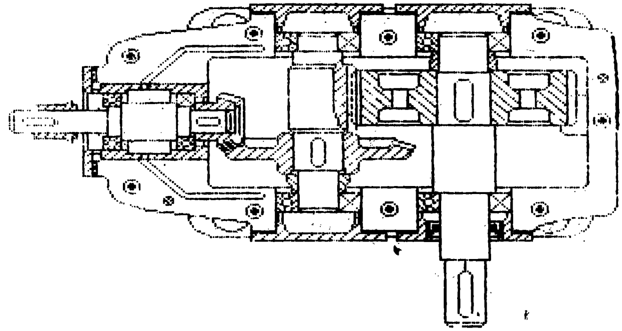


图 1.4 二级圆锥圆柱齿轮减速器

盘，数字化板、鼠标器等对图形内容、格式、大小和色彩在显示屏幕上实现动态控制，最终生成一幅图形。以后章节谈到的交互绘图，输入设备为键盘，输出设备是显示器或打字机的硬拷贝。

第二章　点线法交互程序

本章介绍的系列程序，可达到下述要求：

·友好的人机界面。

·可交互地生成任意方向的直线、圆、圆弧。

·能实现对图形的放大、缩小、旋转、移动。

·能保存各子图形的有关参数。

针对上述各项要求，我们分节介绍程序的编制思想，并给出相应的程序。

第一节　友好的人机界面

① 友好的人机界面准则

衡量一个交互程序的成功与失败，是否方便使用，至少和它的功能一样重要。

在写交互程序时，应考虑友好的人机界面。其遵循的一般原则为：

·提供简单的、前后一致的交互作用顺序。

·为了与程序进行通讯，不要过多地选择方案和格式来加重用户的负担。

·要有提示。

·有反馈信息，对用户错误的输入程序应有坚固性要求。

在下面涉及的交互程序实例中，我们试图遵循上面的准

则，具体采用的方式有图形及其汉字提示，加求助命令。告诉每个命令的具体含义，程序按照层次模块结构。在本篇的最后一章，将给出通用造型软件包的设计思想及总体结构。程序GP 1201是一个主控程序及相应的菜单命令。

② 主控程序

```
5   REM   GP1201
10  HCOLOR= 3
15  HOME : GOSUB 100
20  VTAB 22: INPUT "Enter selection:"
     ;A$
25 A =  VAL (A$)
30  IF A <  = 0 OR A > 8 THEN  CALL 6
     5338: GOTO 20
40  ON A GOSUB 500,600,700,800,900,10
     00,1100
45  IF A = 8 THEN  TEXT : END
50  GOTO 15
100  VTAB 2
105  PRINT "*************************"

110  PRINT "1--Draw a line."
115  PRINT "2--Draw a circle."
120  PRINT "3--Draw a arc."
125  PRINT "4--Zoom a drawing."
130  PRINT "5--Move a drawing."
135  PRINT "6--Save a file to disk."
140  PRINT "7--Help."
145  PRINT "8--End.
150  PRINT "*************************"

160  RETURN
500  REM  Draw a line
595  RETURN
600  REM   Draw a circle
695  RETURN
```

```
700   REM  Draw a arc
795   RETURN
800   REM  Zoom a drawing
895   RETURN
900   REM  Move a drawing
995   RETURN
1000   REM  save a file to disk.
1095   RETURN
1100   REM  Help
1195   RETURN
```

③ 程序说明

程序GP1201的30句的作用为：当用户输入不是屏幕上的选择项时，机器鸣响，要求重新输入选择项。

100—160句为命令菜单子程序，在进入该项程序运行时，屏幕显示该命令菜单，其含义为：

- 绘制直线
- 绘一个圆
- 绘圆弧
- 缩放一幅图形
- 移动一幅图形
- 求助命令
- 存盘
- 结束

程序在进入求助命令模块，按照交互程序设计准则。屏幕上最好有图形提示及其具体操作的细节说明。

第二节　直线的交互实现程序

(1) 程序达到的功能

本程序要求用户在屏幕上用拖动的光标输入两个坐标点，

然后显示这二点的连线，有关节点数据保存在数组里。

② 源程序（见G P 1202）

```
4   REM   GP1202
5   DIM A%(500),B%(500),C%(500),D%(500
       ),E%(500)
6 C1 = 10:X = 100:Y = 80:ARS = 1:HC =
     3
7   HGR : POKE  - 16302,0
40   HCOLOR= 3: HPLOT X,Y
100   GET A$: HCOLOR= 0: HPLOT X,Y
103 E%(J1) = HC
105   IF A$ = "A" THEN  GOSUB 700
110   IF A$ = "I" THEN Y = Y - C1
120   IF A$ = "J" THEN X = X - C1
130   IF A$ = "L" THEN X = X + C1
140   IF A$ = "M" THEN Y = Y + C1
150   IF A$ = "N" THEN  GOSUB 600
155   IF A$ = "C" THEN  GOSUB 750
160   IF A$ = "F" THEN C1 = C1 + 1
162   GOSUB 300
170   IF A$ = "" THEN C1 = 1
190   IF A$ =  CHR$ (13) THEN  CALL 65
      338: GOSUB 230
210   HCOLOR= 3: HPLOT X,Y
220   GOTO 100
225   END
230   IF ARS = 1 THEN A%(J1) = X:B%(J1
      ) = Y:ARS = 2: RETURN
240   IF ARS = 2 THEN C%(J1) = X:D%(J1
      ) = Y:ARS = 1: GOSUB 400:J1 = J1
       + 1: RETURN
250   RETURN
300   IF X < 0 THEN X = 0
310   IF X > 255 THEN X = 255
320   IF Y < 0 THEN Y = 0
330   IF Y > 191 THEN Y = 191
```

```
340  RETURN
400  HCOLOR= E%(J1)
420  HPLOT A%(J1),B%(J1) TO C%(J1),D%
     (J1)
450  RETURN
600  IF J1 < 1 THEN  CALL 65338: GOTO
     620
605 J1 = J1 - 1
610  HCOLOR= 0: HPLOT A%(J1),B%(J1) TO
     C%(J1),D%(J1)
620  RETURN
700  FOR I = 0 TO J1 - 1
710  HCOLOR= E%(I)
720  HPLOT A%(I),B%(I) TO C%(I),D%(I)

730  NEXT
740  RETURN
750  HOME : HGR
755  VTAB 24: INPUT "Color=";HC
758  IF HC < 0 OR HC > 7 THEN  HOME :
      GOTO 755
760  POKE   16302,0: RETURN
```

(3) 程序说明

本程序绘图的起点坐标为（100,80)，在第一页画图。共设置了A%(I)，B%(I)，C%（I)，D%（I)，E%（I）5个数组，分别存放直线的起点坐标，终点坐标以及该直线的颜色。开始绘图时，颜色号码设定为3。

4—225为主程序

A%(I)、B%(I) 数组存放直线的起点坐标，C%(I)、D%(I) 数组存放直线的终点坐标值，E%(I) 数组存贮颜色。

100～210两句可控制屏幕上一个动态光点，让用户观察

坐标的位置。按“A”键将存贮在数组的数据作为图形数据重新显示一遍。按“I”键使光标上移，按“J”键使光标左移，按“L”键使光标右移，按“M”键使光标下移，按“N”键为删除当前画的直线，按“C”键为改变要绘直线的颜色，按“CTRL-F”键为改光标移动的距离，当按下此键后，再按“I”、“J”、“L”及“M”方向键时，光标只移动一个坐标单位。按“F”键为移动单位增加，每按一次，则增加一个坐标单位，按回车键表示选中该点的坐标值，并将其送入数组中，同时在屏幕上显示这条直线。

300—340子程序为控制光标不超出屏幕以外而设置。

600—620为删除子程序，按“N”键进入该子程序，删除数组中的数据，同时将屏幕上的线条抹去。

700—740子程序为显示图形模块。

750—760为改变颜色子程序。

本程序存贮在数组中的数据为各条直线的始点及终点，也称这样的点为“节点”。人与屏幕的交互作用，实际上是对数组中数据的操作。

第三节 圆的交互程序

本程序要求在屏幕上指出两点，圆心和半径，能绘制任意不超过屏幕边界的圆。有关圆周上坐标点的数据，只要计算出0到π/4范围中各点的坐标，根据圆的对称性来推算其他坐标，从而大大加快画圆的速度。

① 源程序（见GP1203）

```
5   REM    GP1203
10 C1 = 10:X = 100:Y = 80:ARS = 1:HC =
```

```
    3
15  HGR2
20 PI = 3.14159
30 F = PI / 180
40  HCOLOR= 3: HPLOT X,Y
100  GET A$: HCOLOR= 0: HPLOT X,Y
110  IF A$ = "I" THEN Y = Y - C1
120  IF A$ = "J" THEN X = X - C1
130  IF A$ = "L" THEN X = X + C1
140  IF A$ = "M" THEN Y = Y + C1
160  IF A$ = "F" THEN C1 = C1 + 1
162  GOSUB 200
170  IF A$ = "" THEN C1 = 1
175  IF A$ =  CHR$ (13) THEN  CALL 65
     338: GOSUB 230
180  HCOLOR= 3: HPLOT X,Y
185  GOTO 100
190  END
200  IF X < 0 THEN X = 0
205  IF X > 255 THEN X = 255
207  IF Y < 0 THEN Y = 0
208  IF Y > 191 THEN Y = 191
210  RETURN
230  IF ARS = 1 THEN A%(J1) = X:B%(J1
     ) = Y:ARS = 2:: RETURN
231  IF ARS = 2 THEN C%(J1) = X:D%(J1
     ) = Y:ARS = 1: GOSUB 233:J1 = J1
      + 1: RETURN
233 A1 = A%(J1):B1 = B%(J1):A2 = C%(J
     1):B2 = D%(J1)
234 R =  SQR ((A1 - A2) * (A1 - A2) +
     (B1 - B2) * (B1 - B2))
235  IF R = 0 THEN J1 = J1   1: RETURN

240 DA = 1 / R:T45 = 45 * F
245  FOR T = 0 TO T45 STEP DA
250 DX = R *  COS (T)
```

```
255 DY = R * SIN (T)
260  HCOLOR= 3: GOSUB 300
267  NEXT T
270  RETURN
300  HPLOT X + DX,Y + DY
305  HPLOT X - DX,Y + DY
310  HPLOT X + DX,Y - DY
315  HPLOT X - DX,Y - DY
320  HPLOT X + DY,Y + DX
325  HPLOT X - DY,Y + DX
330  HPLOT X + DY,Y - DX
340  HPLOT X - DY,Y - DX
350  RETURN
```

② 程序说明

5—190为主程序，40与180两句实现一个拖动的光点。按回车键表示接收屏幕上的坐标点。

233—270子程序的作用，计算半径、0—π/4范围内圆周的坐标点。234句由接收的两个坐标点计算圆的半径。235句为当半径为零，实际上是在同一位置上输入两个点时，退出并取消这两点的坐标。

300—350子程序的作用，根据计算的0—π/4范围的坐标点，由圆的对称性推算圆周上其他7个坐标点，若已计算出（X，Y），则与它对称的圆周上另外7个坐标点分别是（X，−Y），（−X，Y），（−X，−Y），（Y，X），（Y，−X）、（−Y，X）和（−Y，−X）。

第四节　圆弧的交互实现程序

① 程序要求

在屏幕上任意输入三点，圆心、圆弧的起始点以及方向，可绘出不超过边界的任一圆弧。有显示功能。

源程序（见GP1204）

```
5  REM   GP1204
6  C1 = 10:X = 100:Y = 80:ARS = 1:HC =
      3
7  HGR : POKE   16302,0
40  HCOLOR= 3: HPLOT X,Y
100  GET A$: HCOLOR= 0: HPLOT X,Y
105  IF A$ = "A" THEN  GOSUB 700
110  IF A$ = "I" THEN Y = Y - C1
120  IF A$ = "J" THEN X = X - C1
130  IF A$ = "L" THEN X = X + C1
140  IF A$ = "M" THEN Y = Y + C1
160  IF A$ = "F" THEN C1 = C1 + 1
162  GOSUB 300
170  IF A$ = "" THEN C1 = 1
190  IF A$ =  CHR$ (13) THEN  CALL 65
     338: GOSUB 230
210  HCOLOR= 3: HPLOT X,Y
220  GOTO 100
225  END
230  IF ARS = 2 THEN A%(J1) = X:B%(J1
     ) = Y:ARS = 3: RETURN
231  IF ARS = 1 THEN C%(J1) = X:D%(J1
     ) = Y:ARS = 2: RETURN
232  IF ARS = 3 THEN E%(J1) = X:F%(J1
     ) = Y:ARS = 1: GOSUB 233:J1 = J1
      + 1: RETURN
233 A1 = A%(J1):B1 = B%(J1):A2 = C%(J
     1):B2 = D%(J1):A3 = E%(J1):B3 =
     F%(J1)
234 R =  SQR ((A1 - A2) * (A1 - A2) +
     (B1 - B2) * (B1 - B2)):R1 = R:R2
      = R
235 AI = A2 - A1:BI = B2 - B1:AK = A3
      - A1:BK = B3 - B1
236 P = 3.14159
237  IF AI = 0 AND BI < 0 THEN F0 = P
```

```
     / 2: GOTO 242
238  IF AI = 0 AND BI > 0 THEN FO = P
     * (3 / 2): GOTO 242
239  IF BI = 0 AND AI > 0 THEN FO = 0
     : GOTO 242
240  IF BI = 0 AND AI < 0 THEN FO = P
     : GOTO 242
241 FO =  ATN ( ABS (BI / AI))
242  IF AK = 0 AND BK < 0 THEN FK = P
     / 2: GOTO 247
243  IF AK = 0 AND BK > 0 THEN FK = P
     * (3 / 2): GOTO 247
244  IF BK = 0 AND AK > 0 THEN FK = 0
     : GOTO 247
245  IF BK = 0 AND AK < 0 THEN FK = P
     : GOTO 247
246 FK =  ATN ( ABS (BK / AK))
247  IF BI < 0 AND AI > 0 THEN FO = F
     O

248  IF BI > 0 AND AI < 0 THEN FO = 1
     80 * (P / 180) + FO
249  IF BI > 0 AND AI > 0 THEN FO = 3
     60 * (P / 180) - FO
250  IF BI < 0 AND AI < 0 THEN FO = 1
     80 * (P / 180) - FO
251  IF BK > 0 AND AK > 0 THEN FK = 3
     60 * (P / 180) - FK
252  IF BK < 0 AND AK < 0 THEN FK = 1
     80 * (P / 180) + FK
253  IF BK < 0 AND AK > 0 THEN FK = F
     K
254  IF BK > 0 AND AK < 0 THEN FK = F
     K + 180 * (P / 180)
255 IO = (FO * 180) / 3.14159:I1 = (F
     K * 180) / 3.14159
256  IF IO > I1 THEN IO = IO - 360
257  IF  INT (R1) = 0 THEN J1 = J1 -
```

```
       1: CALL 65338: RETURN
258 C =  INT (50 / R1 + 1)
259 F = 3.14159 / 180
260 A%(J1) = A1:B%(J1) = B1:C%(J1) =
       R1:D%(J1) = R2:E%(J1) = I0:F%(J1
       ) = I1
262  FOR I = I0 TO I1 STEP C
263 X = A1 + R1 *  COS (F * I)
264 Y = B1 - R2 *  SIN (F * I)
265 A% =  INT (X + .5):B% =  INT (Y +
       .5)
266  HCOLOR= 3: HPLOT A%,B%
267  NEXT I
268  RETURN
300  IF X < 0 THEN X = 0
305  IF X > 255 THEN X = 255
310  IF Y < 0 THEN Y = 0
315  IF Y > 191 THEN Y = 191
320  RETURN
```

③ 程序说明

程序共设置了A%（I）、B%(I)、C%（I）、D%(I)、E%(I)和F%(I)数组，分别存贮圆心坐标，半径及圆弧的起始弧度、终止弧度。角度的定义逆时针为正。

5—255为主程序。100与210两句能实现一个拖动的光点。

230—232子程序的作用，顺序接收屏幕上坐标点，第一次输入为圆心坐标，第二次输入为弧段起始位置坐标值，第三次输入为弧段的终止方向，并分别存入有关数组里。

233—268子程序的作用，根据接收的三点，计算出圆弧的半径；根据象限的判定及边界处理来确定起始弧度值和终止弧度值，并按逆时针方向显示该圆弧。起始弧度与终止弧

度大小的确定见图1.5,坐标原点设在圆心。主要根据X、Y的正负，来确定Q值的大小。即：

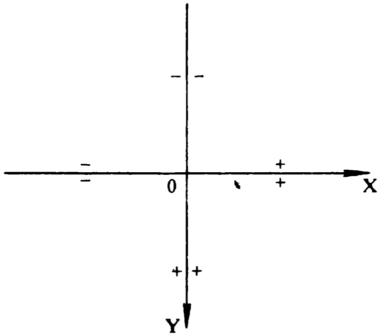


图 1.5 Q值大小的确定

当X为正，Y为负，则$Q=\text{arctg}\left(\frac{|Y|}{X}\right)$。

当X为负，Y为负，则$Q=90°+\text{arctg}\left(\frac{|Y|}{|X|}\right)$。

当X为负，Y为正，则$Q=180°+\text{arctg}\left(\frac{Y}{|X|}\right)$。

当X为正，Y为负，则$Q=270°+\text{arctg}\left(\frac{|Y|}{X}\right)$。

第五节 图形的二维变换交互程序

本节主要介绍计算机图形的基本的二维变换：平移、比例变换及旋转、并给出相应的交互程序。

1．平移变换

把图形从屏幕上的一处移动到另一处，称为图形的“平

```
5 REM GP1205
10 HOME : HGR2
20 X = 100:Y = 80:A = 25:B = 10:ME
      = 10
30 C = 3: GOSUB 300
100 GET A$
102 C = 0: GOSUB 300
110 IF A$ = "I" THEN Y = Y - ME
120 IF A$ = "J" THEN X = X - ME
130 IF A$ = "M" THEN Y = Y + ME
140 IF A$ = "L" THEN X = X + ME
150 IF A$ = "F" THEN ME = ME + 1
160 IF A$ = "S" THEN ME = 1
165 IF A$ = "E" THEN TEXT : END

170 C = 3: GOSUB 300
180 GOTO 100
200 END
300 HCOLOR= C
305 HPLOT X,Y TO X + A,Y TO X + A
    ,Y + B
310 HPLOT X,Y + B TO X + A,Y + B
320 HPLOT X,Y TO X,Y + A / 2 + B
330 RETURN
```

移”，假设图上某点的老坐标为（X 1, Y 1），平 移 后坐标为（X，Y），平移的水平距离为X 0，垂直距离为Y 0，则：

$$\begin{cases} X = X0 + X1 \\ Y = Y0 + Y1 \end{cases}$$

当X 0为正表示向右移动，X 0为负则向左移动，Y 0为正则表示向下移动，Y 0为负则表示向上移动。

对屏幕上显示的图形，要移动该幅图形，只要做到下列点：

·计算被平移图形中节点的新坐标。

·擦去老图形。

·按新坐标位置显示这幅图形。

下面的程序可实现一个小旗的移动，见程序ＧＰ 1205。

① 源程序

② 程序说明

5—200为主程序，102与170两句可实现小旗的拖动。

300—330子程序为画小旗的模块。

对圆及圆弧我们只要变换相应的节点——圆心，相应程序可仿照上述的程序来实现。不过圆的拖动用节点法在中华学习机上，由于速度较慢，实现这一要求速度不允许。

需要注意一点，在变换过程中，要考虑图形是否画到边界以外，这方面要增加检验子程序，以免画到屏幕以外而造成程序执行中止。

2．图形的比例变换

所谓比例变换，是将图形沿Ｘ轴方向扩大（或缩小）S_X倍，沿Ｙ轴方向扩大（或缩小）S_Y倍，见图1.6。

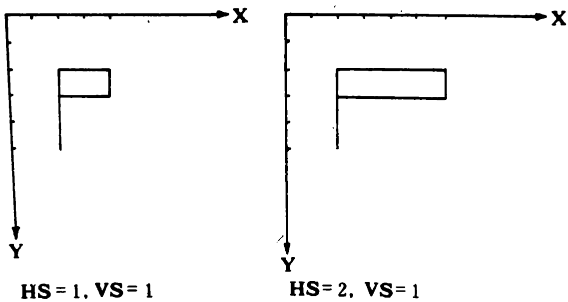


图 1.6　比例交换的定义

当 $S_X = S_Y$ 时，被变换图形只改变大小而其形状不改变。

进行比较变换时，如相对于屏幕上的某点（X0，Y0）进行变换，其变换公式如下：

$$\begin{cases} X_s = X_0 + (X - X_0) * S_x \\ Y_s = Y_0 + (Y - Y_0) * S_y \end{cases}$$

下面的程序可实现上述小旗的缩放，见GP 1206。

① 源程序

```
5  REM   GP1206
10  HOME
20 X = 100:Y = 80:A = 25:B = 10:ME
      = 10
50  VTAB 24: INPUT "Enter scale-X"
     ;SX
60  INPUT "Enter scale-Y";SY
75 A = A * SX:B = B * SY
80  HGR2 :C = 3: GOSUB 300
100  GET A$
102 C = 0: GOSUB 300
110  IF A$ = "I" THEN Y = Y    ME
120  IF A$ = "J" THEN X = X - ME
130  IF A$ = "M" THEN Y = Y + ME
140  IF A$ = "L" THEN X = X + ME
150  IF A$ = "F" THEN ME = ME + 1
160  IF A$ = "S" THEN ME = 1
165  IF A$ = "E" THEN  TEXT : END

170 C = 3: GOSUB 300
180  GOTO 100
200  END
300  HCOLOR= C
305  HPLOT X,Y TO X + A,Y TO X + A
     ,Y + B
310  HPLOT X,Y + B TO X + A,Y + B
320  HPLOT X,Y TO X,Y + A / 2 + B
330  RETURN
```

② 程序说明

10—200句为主程序，50—60句让用户顺序输入沿X轴和沿Y轴的缩放比例S_X和X_Y，75句为小旗相对于左端点(100，80）的缩放。

3．图形的旋转

所谓"旋转变换"是将屏幕上显示的图形绕某一点旋转一定的角度，见图1.7。

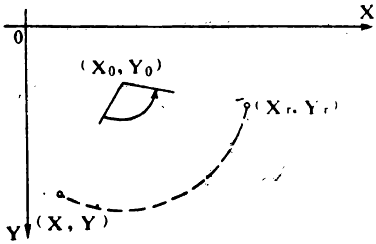


图 1.7　点的旋转变换

基准点为（X_0，Y_0）逆时针旋转了T角度，其变换公式如下：

$$X_r = X_0 + (X - X_0) * \cos(T) + (Y - Y_0) * SIN(T)$$

$$Y_r = Y_0 + (Y - Y_0) * \cos(T) - (X - X_0) * SIN(T)$$

下面的程序可对上述的小旗进行旋转变换，见GP 1207。

① 源程序

```
5   REM    GP1207
10   HOME
20 X = 100:Y = 80:A = 25:B = 10:ME =
```

```
     10
50  VTAB 24: INPUT "Enter ST:";ST
55 T = 3.14159 / 180 * ST
60 A1 = A *  COS (T) + 0 *  SIN (T):
      B1 = 0 *  COS (T) - A *  SIN (T
      )
65 A2 = A *  COS (T) + B *  SIN (T):
      B2 = B *  COS (T) - A *  SIN (T
      )
70 A3 = 0 *  COS (T) + B *  SIN (T):
      B3 = B *  COS (T) - 0 *  SIN (T
      )
80 A4 = 0 *  COS (T) + (B + A / 2) *
       SIN (T):B4 = (B + A / 2) *  COS
      (T) - 0 *  SIN (T)
90  HGR2 :C = 3: GOSUB 300
100  GET A$
102 C = 0: GOSUB 300
110  IF A$ = "I" THEN Y = Y - ME
120  IF A$ = "J" THEN X = X - ME
130  IF A$ = "M" THEN Y = Y + ME
140  IF A$ = "L" THEN X = X + ME
150  IF A$ = "F" THEN ME = ME + 1
160  IF A$ = "S" THEN ME = 1
165  IF A$ = "E" THEN  TEXT : END
170 C = 3: GOSUB 300
180  GOTO 100
200  END
300  HCOLOR= C
305  HPLOT X,Y TO X + A1,Y + B1
308  HPLOT X + A1,Y + B1 TO X + A2,Y
      + B2
310  HPLOT X + A2,Y + B2 TO X + A3,Y
      + B3
320  HPLOT X,Y TO X + A4,Y + B4
330  RETURN
```

② 程序说明

5—200为主程序，50句可让用户输入逆时针旋转的角度，60—90句为计算小旗各端点旋转后相对于左端点（100，80）的坐标。

300—330子程序为画出旋转后小旗的模块。

4．交互绘图的变换

以上介绍了几种常用的变换，并以小旗图块为例给出了几个实例程序。只要对小旗图块子程序稍加变动，或者用其它图块替代小旗图块，能实现不同图块的二维变换。但在交互程序中，如何对已经作好的图形施加上述的变换呢？我们知道，交互程序画图存贮的是图形的有关节点——直线的起始点、终止点。圆的圆心、半径，圆弧的圆心、半径、起始角和终止角。根据上述原理，只要对这些节点施加变换就能达到要求，下面将给出交互画直线，并给出上述几种变换的程序，见GP 1208。

① 源程序

```
5   REM   GP1208
6   DIM A%(100),B%(100),C%(100),D%(1
      00),E%(100)
10   HGR :J1 = 1:ME = 10:ARS = 1
15  X = 80:Y = 80
20   VTAB 24: INPUT "Command:";A$
25   IF A$ = "MOVE" THEN  GOSUB 100
30   IF A$ = "ZOOM" THEN  GOSUB 800
40   IF A$ = "ROT" THEN  GOSUB 900
50   IF A$ = "LINE" THEN  GOSUB 200
60   IF A$ = "END" THEN  TEXT : END
70   GOTO 20
90   END
```

```
100  GET A$: HCOLOR= 0: HPLOT X,Y
105  IF A$ = "I" THEN Y = Y - ME
110  IF A$ = "J" THEN X = X - ME
120  IF A$ = "M" THEN Y = Y + ME
125  IF A$ = "L" THEN X = X + ME
130  IF A$ = "F" THEN ME = ME + 1
140  IF A$ = "S" THEN ME = 1
145  IF A$ = "Q" THEN  RETURN
150  IF X = 0 THEN  RETURN
160  IF A$ =  CHR$ (13) THEN  GOSUB
     750
170  HCOLOR= 3: HPLOT X,Y
180  GOTO 100
200  GET A$: HCOLOR= 0: HPLOT X,Y
210  IF A$ = "I" THEN Y = Y - ME
220  IF A$ = "J" THEN X = X - ME
230  IF A$ = "M" THEN Y = Y + ME
240  IF A$ = "L" THEN X = X + ME
250  IF A$ = "F" THEN ME = ME + 1
255  IF A$ = "Q" THEN  RETURN
260  IF A$ = "S" THEN ME = 1
270  IF A$ =  CHR$ (13) THEN  CALL
     65338: GOSUB 300
275  HCOLOR= 3: HPLOT X,Y
280  GOTO 200
300  IF ARS = 1 THEN A%(J1) = X:B%(
     J1) = Y:ARS = 2: RETURN
310  IF ARS = 2 THEN C%(J1) = X:D%(
     J1) = Y:ARS = 1: GOSUB 400:J1 =
     J1 + 1: RETURN
320  RETURN
400  HCOLOR= 3: HPLOT A%(J1),B%(J1)
      TO C%(J1),D%(J1)
420  RETURN
600  IF J1 < 1 THEN  CALL 65334: GOTO
     630
610 J1 = J1 - 1
```

```
620  HCOLOR= 0: HPLOT A%(J1),B%(J1)
      TO C%(J1),D%(J1)
630  RETURN
700  FOR I = 1 TO J1 - 1
720  HPLOT A%(I),B%(I) TO C%(I),D%(
     I)
730  NEXT
740  RETURN
750  IF ARS = 1 THEN X0 = X:Y0 = Y:
     ARS = 2: RETURN
755  IF ARS = 2 THEN MX = X - X0:MY
      = Y - Y0:ARS = 1
760  HCOLOR= 0: GOSUB 700
765  FOR I = 1 TO J1 - 1
770 A%(I) = A%(I) + MX:B%(I) = B%(I
     ) + MY
775 C%(I) = C%(I) + MX:D%(I) = D%(I
     ) + MY
777  NEXT
780  HCOLOR= 3: GOSUB 700
785  RETURN
800  VTAB 24: INPUT "Enter scale-X:
     ";SX
810  INPUT "Enter scale-y:";SY
815  INPUT "Enter the base of point
      x,y:";X0,Y0
820  HCOLOR= 0: GOSUB 700
825  FOR I = 1 TO J1 - 1
830 A%(I) = X0 + (A%(I) - X0) * SX
835 B%(I) = Y0 + (B%(I) - Y0) * SY
840 C%(I) = X0 + (C%(I) - X0) * SX
845 D%(I) = Y0 + (D%(I) - Y0) * SY
847  NEXT I
850  HCOLOR= 3: GOSUB 700
860  RETURN
900  VTAB 24: INPUT "Enter thre:";T
```

```
910  INPUT "Enter base of x,y:";X0,
     Y0
915 T = 3.14159 / 180 * T
920  HCOLOR= 0: GOSUB 700
930  FOR I = 1 TO J1 - 1
935 X1 = X0 + (A%(I) - X0) * COS (
     T) + (B%(I) - Y0) * SIN (T)
940 Y1 = Y0 + (B%(I) - Y0) * COS (
     T) - (A%(I) - X0) * SIN (T)
945 X2 = X0 + (C%(I) - X0) * COS (
     T) + (D%(I) - Y0) * SIN (T)
950 Y2 = Y0 + (D%(I) - Y0) * COS (
     T) - (C%(I) - X0) * SIN (T)
955 A%(I) = X1:B%(I) = Y1:C%(I) = X
     2:D%(I) = Y2
960  NEXT I
965  HCOLOR= 3: GOSUB 700
970  RETURN
```

② 程序说明

5—90为主程序，按MOVE回车进入移形模块，按ZOOM回车进入缩放模块，按ROT回车进入旋转模块，按LINE回车进入交互绘直线模块，按END回车结束程序运行。

200—280子程序为绘直线模块，其含义见GP 1202程序的说明。

100—180子程序为移形模块，其含义类似于GP 1205程序说明。

800—860子程序为缩放模块，进入该模块后，要求用户输入沿X轴与Y轴的缩放比例 S_X与S_Y，还要输入被缩放图形的基准点坐标（X_0，Y_0），825—847 为将图形各节点相对于基准点（X_0，Y_0）进行的缩放。850 句将缩放后的图

形显示在屏幕上。

900—970子程序为旋转模块，进入该模块后，要求用户顺序输入旋转角度以及被旋转图形的基准点坐标（X_0, Y_0），930—955为将图形各节点相对于基准点（X_0，Y_0）进行的旋转。

第六节　图形存贮

以上我们讨论了在中华学习机上如何实现交互绘图并给出了相应的程序。下面谈谈如何将高解析度画面的图形存贮在软盘上（即外存贮器）。

一、印象存贮

中华学习机有高解析度图形两页，第一页的图形缓冲区十六进制为＄2000—＄3FFF，第二页为＄4000—＄5FFF。屏幕上显示的图形信息与相应的缓冲区内存具有对应关系，我们只要将其缓冲区信息存贮（印象存贮）下来，就能保存这幅图形，假设名字为B＄=“NAME”,则：

在第一页，可用命令：

```
] BSAVE  B＄, A＄2000, L＄2000
```

在第二页，可用命令：

```
] BSAVE  B＄, A＄4000, L＄2000
```

上述两条命令，可将相应的图形存贮在软盘上。要想显示这幅图形，在进入图形状态后，用下列命令即可从软盘上将图形调出并显示。其命令如下：

```
] HGR2
] BLOAD  B＄, A＄4000
```

用BSAVE与BLOAD命令，DOS3.3操作系统必须

事先装入内存，它们也能带行号插在程序之中，例如：

```
5   HGR 2
10  PRINT CHR$(4);"BLOAD";
    B$;", A$4000"
    ⋮
```

二、用数据文件存贮图形

用印象存贮一幅图形十分简单，但图形数据占用内存较大，且调盘的速度慢，建立好的图形，下次使用时不便于插入与增加，一般情况下，我们不使用这种方法来存贮图形，下面介绍另一种存贮图形数据的方法。

以上谈到用“节点法”实现的交互绘图程序，其图形存贮的是有关节点，并且将它们保存在数组里，但一旦程序停止运行，关机以后将会失掉这些数据，如果我们能将这些数据保存在外存贮器中，那么也就保存了这幅图形。保存数据常用的方法是用数据文件。下面我们就介绍用数据文件来保存图形数据的方法。

以直线交互实现程序为例，建立一个存贮图形有关节点的随机文件和一个读这幅图形数据的随机文件。

1. 建立数据文件保存直线图形数据的程序

①源程序（见GP1209）

```
5   REM   GP1209
800  TEXT : HOME : VTAB 24: INPUT "
     Enter NAME---";B$
810  REM  SET TEXT OF DRAWING
820 D$ =  CHR$ (4)
830  PRINT D$;"OPEN";B$;",L20"
840  PRINT D$
845  PRINT D$;"WRITE";B$;",R0"
```

```
848  PRINT J1
849  FOR I = 1 TO J1
850  PRINT D$;"WRITE";B$;",R";I
870  PRINT A%(I - 1);",";B%(I - 1);
     ",";C%(I - 1);",";D%(I - 1);",
     ";E%(I - 1)
880  NEXT I
890  PRINT D$;"CLOSE"
900  HGR : POKE - 16302,0: RETURN
```

②程序说明

文件名让用户输入，比如为DR，每个记录的长度为20，分别存贮每条直线的颜色，起点坐标与终点坐标，零号记录存贮屏幕上显示的直线数目。

830语句为打开名为DR的随机文件，其记录长度为20。

845—848行是将J1（直线数目)写到零号记录上。

849—980行是将A%(I)、B%(I)、C%(I)、D%(I)、E%(I)数据存贮在i（1,2,…，J1）记录上，每个记录存贮一条直线的5个节点数据，起点坐标、终点坐标以及颜色。

890句为关闭这个随机文件。

该程序使用必须加在直线的交互程序GP 1202中，在主程序中加一条语句以表示进入这个模块，该语句为：

165 IF A$="S" THEN GOSUB 800

2．读上述随机文件的程序

① 源程序（见GP 1210）

```
5   REM    CP1210
6   DIM A%(500),B%(500),C%(500),D%(
       500),E%(500)
```

```
8 B$ = "DR"
10  REM  READ TEXT OF DRAWING
20 D$ =  CHR$ (4)
30  PRINT B$;"OPEN";B$;",L20"
40  PRINT D$
42  PRINT D$;"READ";B$;",R0"
43  INPUT J1
45  FOR I = 1 TO J1
60  PRINT D$;"READ";B$;",R";I
75  INPUT A%(I),B%(I),C%(I),D%(I),
     E%(I)
80  NEXT I
90  PRINT D$;"CLOSE"
92  FOR I = 1 TO J1
93  PRINT A%(I);",";B%(I);",";C%(I
     );",";D%(I);",";E%(I)
98  NEXT
100  HGR : POKE  16302,0
110  HCOLOR= 3
130  FOR I = 1 TO J1
150  HCOLOR= E%(I)
160  HPLOT A%(I),B%(I) TO C%(I),D%
     (I)
180  NEXT
200  END
```

② 程序说明

10—90行为读名为DR的随机文件的程序，将数据分别读进J1变量以及A%(I)、B%(I)、C%(I)、D%(I)、E%(I)数组里，与上述建立数据不同的是60语句为读随机文件语句。

92—98行为打印各条直线的起点与终点坐标以及颜色。

100—180行将从数据文件读到的数据作为图形数据将图形显示在屏幕上。

对于圆、圆弧及二维变换后的数据，我们也可以同样设计类似的数据文件格式及相应存贮数据及读文件的程序。

但是，如果把上述几个子程序连成一体，如何设计文件的格式，当打开这样的文件，如何区分哪个记录是直线，哪个记录是圆或圆弧的数据，并按显示程序的要求将这些数据读到相应的数组里。我们可以这样来考虑，设置关字键分别加到直线、圆和圆弧的数据格式之首，并按最长的格式来设定随机文件记录长度。其相应的数据格式如下：

· 对直线，关键字1、起始点坐标、终止点坐标、颜色，则一条直线需要数组中6个数据来存放。

· 对圆，关键字2，圆心坐标、半径、颜色、则一个圆需要数组中5个数据来存放。

· 对圆弧，关键字3，圆心坐标，起始与终止弧度值、颜色，则一个圆弧需要数组中6个数据来存放。

这样，当第一个关键字为圆，下面的数据是一个圆的有关数据。同样，当关键字是直线，则随后的数据为直线数据。圆、圆弧及直线的关键字可任意约定。

按照上述要求设计的程序便于图形的交互处理，如果要删除某一条直线，则只要将相应该记录的关键字置为零。上述的存贮约定及格式，可以增加指针及索引。

第三章　造型法生成工具

在第二章我们介绍了点线法的交互程序设计，并给出了相应程序。众所周知，在中华学习机上，用点线法生成曲线不管是被动式作图，还是交互作图，显示图形速度慢，当显示较多曲线时，显示速度将更成问题。而采用造型法可避免上述问题，但靠手工或者靠辅助工具；生成复杂造型或者函数曲线造型也是十分困难的，本章介绍交互生成造型表的一系列交互工具，可满足任意复杂造型或具有函数关系曲线的造型生成。象圆或者是三角函数曲线的拖动实现将是十分方便的。

第一节　手控造型法的交互生成程序

① 程序要求

把全屏幕当一张纸，用模拟光笔(箭头）来在纸上画图，对画错的图能即时删除。

从第一个造型开始，按顺序可以建立255个造型。

建立好的造型表可以存盘。

② 源程序（见GP 1301）

```
5  REM  GP1301
8  DATA      1,0,4,0,63,45,45,28,63,12,
     6,0
9  FOR I = 0 TO 11
```

```
10  READ A: POKE 768 + I,A: NEXT
12 N = 24576:K1 = 232:K2 = 233
15  SCALE= 1: ROT= 0
20  HOME : INPUT "Enter Numbers of sh
     ape:";M
25  POKE N,M:NU = N + 2 * (M + 4)
30  HGR2 : GOSUB 365: POKE K1,0
210  POKE K2,96: ROT= 0: DRAW J AT X,
     Y
215  GET A$: POKE K2,3: HCOLOR= 0
218  ROT= RT: DRAW 1 AT 200,10
224  IF A$ = "Q" THEN  GOSUB 480: END

230  IF A$ = "E" THEN M% = 0: GOTO 29
     5
235  IF A$ = "S" THEN M% = 3: GOTO 29
     5
240  IF A$ = "F" THEN M% = 1: GOTO 29
     5
245  IF A$ = "D" THEN M% = 2: GOTO 29
     5
260  IF A$ = "B" THEN  GOSUB 405
270  IF A$ = "I" THEN M% = 4: GOTO 29
     5
275  IF A$ = "J" THEN M% = 7: GOTO 29
     5
280  IF A$ = "L" THEN M% = 5: GOTO 29
     5
285  IF A$ = "K" THEN M% = 6: GOTO 29
     5
287  IF A$ = "C" THEN  HGR2 : GOSUB 3
     65
290  GOTO 210
295  HCOLOR= 3:RT = 16 * M%: ROT= RT:
      DRAW 1 AT 200,10
300  HCOLOR= 3:RT = 16 * M%: ROT= RT:
      DRAW 1 AT 200,10
```

```
305  IF T% = 2 AND (M% > 3 OR M% = 0)
      THEN 330
315 S% = S% + 8 ^ T% * M%: POKE NU,S%
     : POKE NU + 1,0
320  IF T% < 2 THEN T% = T% + 1: GOTO
     210
325 NU = NU + 1:T% = 0:S% = 0: GOTO 2
     10
330  IF S% = 0 THEN  POKE NU,128
335 NU = NU + 1:T% = 1:S% = M%: POKE
     NU,S%: POKE NU + 1,0
340  GOTO 210
350  END
365 NU = NU + 1: POKE NU,0:NU = NU +
     1
370 W% = NU - N:J = J + 1: POKE N + 1
     ,J
375 S% = 0:A =  INT (W% / 256):B = W%
      - A * 256
378  POKE N + 2 * J,B: POKE N + 2 * J
      + 1,A
380  POKE NU + 1,0
385  POKE N + 2 * J + 2,0: POKE N + 2
      * J + 3,0
390 X = 100:Y = 100: RETURN
405  IF NU = N + W% AND  PEEK (NU) =
     0 THEN  RETURN
410  POKE K2,96
415 T% = T% - 1: IF T% < 0 THEN NU =
     NU - 1:T% = 2
420  ROT= 0: HCOLOR= 0: DRAW J AT E%,
     F%
425 S% =  PEEK (NU): IF S% = 0 THEN 4
     15
430 R =  INT (S% / 8 ^ T%)
440  IF T% = 2 AND R = 0 THEN T% = T%
      - 1: GOTO 430
```

```
465 S% = S% - R * 8 ^ T%: POKE NU,S%
470  HCOLOR= 3: DRAW J AT E%,F%
475  RETURN
480  TEXT : HOME : VTAB 22
482  INPUT "Enter NAME of shape-table
     :";B$
485  PRINT  CHR$ (4);"BSAVE";B$;",A";
     N;",L";NU - N + 10
490  RETURN
```

③ 程序说明

5－350为主程序。按E键为向上移动一个不画向量。按S键为向左移动一个不画向量。按D键向下移动一个不画向量。按F键向右移动一个不画向量。按B键为删除，按一次删除一个造型矢量，连续地按可由后向前连续地删除造型矢量。按I键向上移动一个画向量，按J键向左移动一个画向量，按M键向下移动一个画向量，按K键向右移动一个画向量，按Q键为存盘及结束命令。

365—390子程序为设置每个造型的表头，即每个造型的索引项，以指出该造型数据应放在内存何处。

405—475子程序是删除子程序，当造型画得不正确时，可即时删除当前的造型矢量，也能连续地由后向前删除造型矢量。

305—335为根据用户控制键来将屏幕上移动矢量转换成相应的3位二进制编码数据，并将该数据存入内存。

造型表的起始位置为＄6000，变量NU始终指向造型表的尾部。

第二节 按参数来生成造型表的交互程序

上一节我们谈到了用点线法来设计的交互程序，这种方法便于图形的变换，但设计好的复杂图形，要想实现动态移动这幅图形，还不能满足要求，由造型组成的复杂图形，能十分方便地移动。下面主要谈谈如何来设计由参数来交互生成造型，比如折线、圆及圆弧的造型，这要做两方面工作，第一步为交互程序模块，由函数来构造相应的显示程序，由HPLOT X，Y语句来实现。第二步还要将屏幕上一边显示的点，一边转换成造型3位二进制编码数据，并将其送入内存。

下面就以交互生成直线然后直接转换造型表的交互程序，之后谈谈圆及圆弧的造型生成。

① 程序要求

能在屏幕上用光标指出两点，然后在屏幕上逐点显示这条直线，一边生成一边转换成造型表。

有存盘功能。

② 源程序（见GP 1302）

```
4  REM   GP1302
5  DIM A%(100),B%(100),C%(100),D%(1
     00),E%(100)
6 N = 24576:K1 = 232:K2 = 233: SCALE=
     1: ROT= 0:J = 0
7  HOME : INPUT "Enter Numbers of s
     hape:";M
9  POKE N,M:NU = N + 2 * (M + 4)
10  HGR : POKE - 16302,0: POKE K1,
     0: POKE K2,96
20  GOSUB 800
```

```
30 C1 = 10:X = 100:Y = 80:ARS = 1:E
      % = X:F% = Y
40  HCOLOR= 3: HPLOT X,Y
100  GET A$: HCOLOR= 0: HPLOT X,Y

110  IF A$ = "I" THEN Y = Y - C1
120  IF A$ = "J" THEN X = X - C1
130  IF A$ = "L" THEN X = X + C1
140  IF A$ = "M" THEN Y = Y + C1
150  IF A$ = "N" THEN  GOSUB 600
155  IF A$ - "C" THEN  GOSUB 800
160  IF A$ = "F" THEN C1 = C1 + 1
162  GOSUB 300
165  IF A$ = "Q" THEN  TEXT : GOSUB
     900: END
170  IF A$ = "" THEN C1 = 1
190  IF A$ =  CHR$ (13) THEN  CALL
     65338: GOSUB 230
210  HCOLOR= 3: HPLOT X,Y
220  GOTO 100
225  END
230  IF ARS = 1 THEN A%(J1) = X:B%(
     J1) = Y:ARS = 2: RETURN
240  IF ARS = 2 THEN C%(J1) = X:D%(
     J1) = Y:ARS = 1: GOSUB 1000:J1
      = J1 + 1: RETURN
250  RETURN
300  IF X < 0 THEN X = 0
310  IF X > 255 THEN X = 255
320  IF Y < 0 THEN Y = 0
330  IF Y > 191 THEN Y = 191
340  RETURN
400  POKE NU,A(0) + A(1) * 8 + A(2)
      * 64:NU = NU + 1
405  POKE NU + 1,0: RETURN
410  IF X% = A% AND Y% = B% THEN S1
```

```
      = 1: RETURN
420  IF S% = 2 AND A(1) = 0 THEN A(
     2) = 1: GOSUB 400:A(0) = 3:S% =
     1
430  IF S% = 2 THEN A(2) = 0: GOSUB
     400:S% = 0
440  IF B% < Y% THEN A(S%) = 4:Y% =
     Y% - 1: GOTO 480
450  RETURN
450  IF A% > X% THEN A(S%) = 5:X% =
     X% + 1: GOTO 480
460  IF B% > Y% THEN A(S%) = 6:Y% =
     Y% + 1: GOTO 480
470 A(S%) = 7:X% = X% - 1
480 S% = S% + 1
490  IF X% = A% AND Y% = B% THEN S1
      = 2: RETURN
500  IF Y% > B% AND S% < 2 THEN A(S
     %) = 0:Y% = Y% - 1: GOTO 550
510  IF Y% > B% THEN A(S%) = 1 + 2 *
     (A% < X%):X% = X% - (A(S%) = 3
     ) + (A(S%) = 1): GOTO 550
520  IF Y% < B% THEN A(S%) = 2:Y% =
     Y% + 1: GOTO 550
530  IF X% > A% THEN A(S%) = 3:X% =
     X% - 1: GOTO 550
540 A(S%) = 1:X% = X% + 1
550 S% = S% + 1: IF S% < 3 THEN 490

560  GOSUB 400:S% = 0: GOTO 490
600  IF J1 < 1 THEN  RETURN
605 J1 = J1 - 1:NU = E%(J1): POKE N
     U + 1,0
610  HCOLOR= 0: HPLOT A%(J1),B%(J1)
      TO C%(J1),D%(J1)
620  RETURN
800  HGR :X% = 0:Y% = 0:A% = 0:B% =
```

```
     0:X = 100:Y = 80
802  HCOLOR= 3: HPLOT X,Y
805 NU = NU + 1: POKE NU,0:NU = NU +
     1
810 W% = NU - N:J = J + 1: POKE N +
     1,J
820 S% = 0:A = INT (W% / 256):B =
     W% - A * 256: POKE N + 2 * J,B
     : POKE N + 2 * J + 1,A
830  POKE NU + 1,0
840  POKE N + 2 * J + 2,0: POKE N +
     2 * J + 3,0
850 X = 100:Y = 100: RETURN
900  TEXT : HOME : VTAB 22
905  INPUT "Enter NAME of shape-tab
     le:";B$
910  PRINT  CHR$ (4);"BSAVE";B$;",A
     ";N;",L";NU - N + 10
915  RETURN
1000 X% = A%(J1):Y% = B%(J1)
1010  IF A%(J1) = C%(J1) AND B%(J1)
      = D%(J1) THEN : RETURN
1020  IF J1 > = 1 THEN X% = C%(J1 -
     1):Y% = D%(J1 - 1):A% = A%(J1)
     :B% = B%(J1): GOSUB 490
1030  IF J1 = 0 AND A% = 0 AND B% =
     0 THEN  POKE A0 + 2 * J,A%(0):
      POKE B0 + 2 * J,B%(0)
1040 X0 = X%:X1 = Y%
1050 X2 = C%(J1):X3 = D%(J1)
1060 E%(J1) = NU: GOSUB 1070: RETURN

1070  IF X0 = X2 THEN 1130
1080 K = (X1 - X3) / (X0 - X2):B =
     X1 - K * X0
1090 RX = X0 - X2:RY = X1 - X3: IF
      ABS (K) > 1 THEN 1150
```

```
1100  FOR Q = X0 TO X2 STEP  SGN (
     RX) *  SQR (RX ^ 2 + RY ^ 2) /
     ( ABS (RX) +  ABS (RY))
1110 B% = K * Q + B:A% = Q
1120  GOSUB 1180: NEXT : RETURN
1130 A% = X0: FOR Q = X1 TO X3 STEP
      SGN (X3 - X1):B% = Q
1140  GOSUB 1180: NEXT : RETURN
1150  FOR Q = X1 TO X3 STEP  SGN (
     RY) *  SQR (RX ^ 2 + RY ^ 2) /
     ( ABS (RX) +  ABS (RY))
1160 A% = (Q   B) / K + 0.5:B% = Q +
     0.5
1170  GOSUB 1180: NEXT : RETURN
1180  HCOLOR= 3: HPLOT A%,B%: GOSUB
     410
1190  RETURN
```

③ 程序说明

4—225为主程序，主程序的操作类似于GP1202程序。

230—250为接收屏幕上输入点的坐标，并将相应的数据存入数组里，J_1为指针变量。

600—620为删除模块，能删除当前或连续地由后向前删除造型直线。一方面抹掉屏幕上显示的线条，另一方面相应缩短造型表的长度。

800—850为继续下一个造型程序，实际上是设置下个造型的索引项。

1000—1190为直线显示程序，它不同于HPLOT语句画的直线，它是根据屏幕上接收输入点坐标，自动地用画点语句逐点描绘一条直线。

410—450为将屏幕上显示的信息自动转换成3位二进制

数据。

400—405子程序将3位二进制数据转换成十进制并送入内存。

490—510为不画向量的转换。

对于圆及圆弧，用前节的生成圆、圆弧相应模块替换1000—1190子程序，改变相应的删除子程序。对于其它曲线的造型生成，可以用节点法原理，按HPLOT X，Y语句设计相应的模块替换掉相应的1000—1190子程序。

第三节 键盘字符交互生成造型的程序

中华学习机在图形状态下显示西文字符有多种方法，比如用HPLOT制作一系列子程序或者用汇编接口将相应字符打印在屏幕上，但均有不便，前者设计麻烦，且占用较多的内存；后者不能在任意位置上标注西文以及旋转字符或者字符上有角标A_2^1等不能满足要求。下面介绍的程序是借用造型表中各个子图块，采用拼积思想来生成键盘字符的任意组合，组合以后仍然是一个造型。

① 程序要求

能完成键盘字符在图形状态下的任意拼积，并有较好的适时效果。

能存盘，能删除即刻显示的字符。

② 源程序

造型表清单（见GP1303-1）。

```
*2000.2580
2000- 7E 00 02 01 0A 01 16 01
2008- 27 01 37 01 47 01 58 01
```

```
2010- 5F 01 6B 01 76 01 90 01
2018- 9C 01 A1 01 AB 01 B0 01
2020- B8 01 C9 01 D2 01 E3 01
2028- F2 01 FF 01 0D 02 1C 02
2030- 28 02 35 02 42 02 49 02
2038- 52 02 5E 02 68 02 76 02
2040- 81 02 91 02 9E 02 AC 02
2048- B8 02 C5 02 D5 02 E1 02
2050- EE 02 FC 02 06 03 10 03
2058- 1E 03 28 03 35 03 42 03
2060- 4F 03 5C 03 6A 03 77 03
2068- 84 03 90 03 9D 03 A8 03
2070- BA 03 C7 03 D2 03 E2 03
2078- F0 03 FF 03 0A 04 1E 04
2080- 29 04 34 04 42 04 4D 04
2088- 58 04 64 04 71 04 7B 04
2090- 87 04 90 04 9B 04 AA 04
2098- B8 04 C1 04 CC 04 DB 04
20A0- E7 04 F1 04 FE 04 0A 05
20A8- 18 05 25 05 00 00 00 00
20B0- 00 00 00 00 00 00 00 00
20B8- 00 00 00 00 00 00 00 00
20C0- 00 00 00 00 00 00 00 00
20C8- 00 00 00 00 00 00 00 00
20D0- 00 00 00 00 00 00 00 00
20D8- 00 00 00 00 00 00 00 00
20E0- 00 00 00 00 00 00 00 00
20E8- 00 00 00 00 00 00 00 00
20F0- 00 00 00 00 00 00 00 00
20F8- 00 00 00 00 00 00 00 00
2100- 00 00 00 00 00 00 00 00
2108- 00 00 32 75 00 24 76 21
2110- A4 92 12 09 11 00 21 3C
2118- 25 3C 25 34 2E 25 34 2E
2120- 37 3E 2D 37 6E 09 00 2D
2128- 65 1C 3F 1C 0C 2D 3D 27
2130- 36 36 36 4C 09 01 00 60
2138- 0C 0C 0C DC 33 27 4D 92
```

2140-	12	2A	3C	4E	09	01	00	20
2148-	0C	0C	E4	17	76	0E	0E	0E
2150-	1C	0C	DC	93	2A	4D	09	00
2158-	20	24	34	36	B6	49	00	09
2160-	1C	1C	24	0C	0C	94	92	4A
2168-	0A	01	00	0C	0C	24	1C	1C
2170-	94	92	4A	51	01	00	61	0C
2178-	0C	0C	DC	DB	0A	0E	0E	0E
2180-	0E	1F	34	26	24	3F	2D	2D
2188-	3F	24	34	36	76	49	01	00
2190-	49	24	3F	2D	2D	3F	24	76
2198-	09	52	09	00	21	F6	0C	4D
21A0-	00	08	08	3D	3F	2D	2D	95
21A8-	09	01	00	5D	09	0B	0B	00
21B0-	0C	0C	0C	0C	94	92	49	00
21B8-	29	65	24	24	1E	1E	1E	1E
21C0-	24	24	0C	2D	95	92	4A	01
21C8-	00	25	24	24	BC	95	12	2A
21D0-	4D	00	2D	2D	3F	3F	24	0C
21D8-	2D	0C	E4	3F	1E	96	52	49
21E0-	09	01	00	A8	2D	0C	E4	2F
21E8-	0C	E4	3F	1E	96	52	49	09
21F0-	01	00	49	24	24	24	17	17
21F8-	17	2E	2D	AD	0A	01	00	A8
2200-	2D	0C	24	1C	3F	27	2C	2D
2208-	AD	92	52	01	00	20	24	0C
2210-	0C	AD	12	3B	BF	52	2D	0C
2218-	A4	52	09	00	64	0C	0C	0C
2220-	3C	3F	B7	92	89	49	09	00
2228-	29	65	E4	0C	E4	3F	1E	AE
2230-	3D	1E	76	49	00	29	0C	0C
2238-	24	E4	3F	1E	76	2D	95	4A
2240-	01	00	0C	08	18	13	55	11
2248-	00	21	BE	0C	24	80	94	52
2250-	09	00	49	1C	1C	1C	0C	0C
2258-	0C	94	52	91	0A	00	28	2D
2260-	25	38	3F	B7	49	89	09	00

```
2268- 0C 0C 0C 1C 1C 1C 94 92
2270- 09 94 09 0A 59 00 22 60
2278- 0C E4 3F 1E 96 4A 09 49
2280- 00 20 0C 35 36 2F 2D 24
2288- 24 24 3F F7 96 4A 49 51
2290- 00 24 24 0C 0C 15 0E 36
2298- 3F 2F 2D 36 4D 00 25 24
22A0- 24 3C 2D 75 F6 2F 0E F6
22A8- 3F 4D 49 00 20 24 64 2D
22B0- 15 96 F2 3F 4F 49 09 00
22B8- 25 24 24 3C 2D 75 36 36
22C0- 1E 7F 49 09 00 24 20 3D
22C8- 27 24 2D 2D 95 92 1A 3F
22D0- 7F 49 49 01 00 24 2C 3D
22D8- 27 24 2D 2D 95 92 0A 01
22E0- 00 20 24 64 2D B5 3A 35
22E8  36 3F 7F 49 49 00 24 2C
22F0- 2D 3F 27 24 4C 09 32 36
22F8- 36 6E 01 00 25 24 24 3C
2300- 2D 95 92 1A 4D 00 A8 2D
2308- 0C 24 24 A4 92 92 49 00
2310- 24 24 24 4D F1 1E 17 0F
2318- 0E 0E 0E 4D 01 00 24 24
2320- 24 96 12 2E 2D 6D 01 00
2328- 24 24 24 15 15 26 0C 0C
2330- 36 36 36 4D 00 24 24 24
2338- 15 15 15 25 24 36 36 36
2340- 4D 00 20 24 24 29 75 36
2348- 36 1E 3F 4F 49 09 00 24
2350- 24 24 2D 75 F6 3F 97 0A
2358- 09 49 01 00 20 24 64 2D
2360- 0E 36 F6 1C 94 3A 4F 0D
2368- 4D 00 24 24 24 2D 75 F6
2370- 3F 0F 0E 0E 0E 4D 00 A8
2378- 2D 0C E4 3F 1C 64 2D 0E
2380- 95 92 09 00 49 24 24 24
```

```
2388- 3F 2D 2D 96 92 09 01 00
2390- 20 24 24 4C 89 36 36 F6
2398- 3F 2D 4D 01 00 49 E4 E4
23A0- 24 4D 31 F6 B6 49 01 00
23A8- 24 24 24 36 36 36 0C 0C
23B0- 15 15 24 24 24 8C 92 52
23B8- 11 00 64 0C 0C 0C FC 1B
23C0- 76 0E 0E 0E 6E 01 00 49
23C8- 24 E4 1C 6C 09 F6 96 4A
23D0- 09 00 64 0C 0C 0C 3C 3F
23D8- 6F 89 92 12 3F 3F 2D 2D
23E0- 49 00 60 2D 25 1C 3F 4F
23E8- 49 92 33 3F 7F 49 49 00
23F0- 24 24 24 36 2E 0C 75 36
23F8- 1E 27 BB 49 49 01 00 20
2400- 64 2D 0E 16 1E 3F 4F 49
2408- 09 00 80 24 0C 6D 24 34
2410- 36 3E 35 3E 35 1B 7F 31
2418- 49 03 80 4C 11 00 20 64
2420- 2D 0E 3E 3F 16 2D 4D 01
2428- 00 21 24 2F 3D 24 0C 15
2430- 96 52 09 00 A8 2D 0C 24
2438- 24 34 E7 BF 76 65 95 52
2440- 09 00 24 24 24 36 2E 0C
2448- 75 36 6E 01 00 25 24 3C
2450- 0C 09 D8 96 92 6D 01 00
2458- E1 0E 2D 0C 24 E4 58 B1
2460- 92 15 09 00 24 24 24 36
2468- 2E 62 0C 94 D2 94 69 01
2470- 00 2D 27 24 24 3C 95 92
2478- 49 0A 00 24 24 75 36 26
2480- 24 0C 0E 36 6E 01 00 24
2488- 24 2E 0C 75 36 6E 01 00
2490- 20 64 2D 0E 36 1E 3F 4F
2498- 49 09 00 24 24 24 2E 0C
24A0- 75 36 1E 27 BB 4A 11 49
24A8- 01 00 49 21 24 24 34 E7
```

```
24B0- F7 36 0E 25 A9 52 09 00
24B8- 24 24 2E 0C 75 96 09 01
24C0- 00 2D 65 1C 3F 1C 0C 2D
24C8- AD 92 09 00 09 20 24 3F
24D0- 2D 2D 3F 24 36 B6 11 0C
24D8- 55 01 00 20 24 36 76 65
24E0- 25 24 36 36 0D 01 00 09
24E8- 1C 1C 24 4D 31 F6 4E 09
24F0- 00 20 24 36 56 0C 74 62
24F8- 24 34 B6 09 01 00 0C 0C
2500- 0C 0C D4 DB 0E 0E 0E 0E
2508- 4D 00 A8 2D 0C 24 3F E7
2510- 24 4D 31 36 B6 09 01 00
2518- 0C 0C 0C 0C 3F 3F 97 12
2520- 29 2D 6D 01 00 2D 00 00
2528- 00 00 00 00 00 00 00 00
2530- 00 00 00 00 00 00 00 00
2538- 00 00 00 00 00 00 00 00
2540- 00 00 00 00 00 00 00 00
2548- 00 00 00 00 00 00 00 00
2550- 00 00 00 00 00 00 00 00
2558- 00 00 00 00 00 00 00 00
2560- 00 00 00 00 00 00 00 00
2568- 00 00 00 00 00 00 00 00
2570- 00 00 00 00 00 00 00 00
2578- 00 00 00 00 00 00 00 00
2580- 00
```

造型表放在＄2000为起始处，各造型按国标码次序排列，几乎包括了键盘的全部字符，一个造型代表一个字符，在］提示符下，输入 CALL-151 回车，在提示符＊下顺序输入上述数据，待完成后，按CTRL-C回车返回到BASIC提示符］，然后输入 BSAVE 名字，A＄2000，L＄526将上述造型数据存盘，程序GP1303的运行需要上面的造型表（GP1303-1）支持。

```
5  REM   GP1303
6  IF  PEEK (8192) = 126 THEN 10
8  PRINT  CHR$ (4);"BLOAD GP1303-1,A$
    2000"
10  DIM B%(200)
20  HOME : INPUT "Enter numbers of sh
    ape:";M%
50  HGR2
55 N = 24576:X = 80:Y = 80
58 NU = N + 2 * M% + 2: POKE N,M%
59  GOSUB 400
60 K1 = 232:K2 = 233: POKE K1,0
70  SCALE= 1: ROT= 0: HCOLOR= 3
80  GET A$
82  IF A$ =  CHR$ (27) THEN : GOSUB 4
    60: TEXT : END
85  IF A$ =  CHR$ (17) THEN  HGR2 :S1
     = 1: GOSUB 400
87  IF A$ =  CHR$ (5) THEN  TEXT :S1
    2: HOME : GOSUB 460: END
90  IF A$ =  CHR$ (32) THEN  POKE NU,
    73:NU = NU + 1: POKE NU,9:NU = N
    U + 1: POKE NU,0
100 E% =  ASC (A$)
110  IF E% >  = 33 AND E% <  = 90 THEN
     J% = E% = 32: GOSUB 150
120  IF A$ =  CHR$ (8) THEN  GOSUB 35
     0
130  GOTO 80
140  END
150 N% = 8192
160  IF J2 < 0 THEN  RETURN
170 B%(J2) = NU:A% =  PEEK (N% + 2 *
     J%)
180 B% =  PEEK (N% + 1 + 2 * J%)
```

```
190 A% = N% + A% + B% * 256
200  IF  PEEK (A%) = 0 THEN A% = A% +
     1
210  FOR I = 0 TO 12000
220 A =  PEEK (A% + I)
230  IF A = 0 THEN 260
240  POKE NU + I,A
250  NEXT I
260 NU = NU + I: POKE NU,0
270 J2 = J2 + 1
280  IF J2 >  = 30 THEN J2 = 0
290  POKE  - 16304,0: POKE  - 16297,0

300  POKE K2,96: DRAW J AT X,Y
310  RETURN
350  IF J2 = 0 THEN J2 = 0: GOTO 370
355  HCOLOR= 0: DRAW J AT X,Y
360  HCOLOR= 3:NU = B%(J2 - 1)
362  POKE NU,0
365  DRAW J AT X,Y:J2 = J2 - 1
370  RETURN
400  IF J >  PEEK (N) THEN  CALL 6533
     8: RETURN
405 NU = NU + 1: POKE NU,0:NU = NU +
     1:J = J + 1
410 W% = NU - N
420 A% =  INT (W% / 256):B% = W% - A%
      * 256: POKE N + 2 * J,B%: POKE
     N + 2 * J + 1,A%
430  POKE NU + 1,0
440  POKE N + 2 * J + 2,0: POKE N + 2
      * J + 3,0
450  RETURN
460  HOME : VTAB 24: INPUT "Enter NAM
     E---";B$
470  PRINT  CHR$ (4);"BSAVE";B$;",A";
```

```
        N;",L";NU - N + 10
480   RETURN
```

③ 程序说明

5—140为主程序，58与59两句为设置造型的表头空间，并建立第一个造型的索引项，按“CTRL-Q”键继续建立下个造型的索引项，并从下个造型开始生成字符造型，90句表示按空格键，屏幕移动一段不画向量，按ESC键进入存盘模块，110句为接受键盘字符，按A—Z键屏幕显示一个相应的字符。

150—310子程序为将$2000处造型表的字符编码按一定格式放入用户造型表之中，180句为取出$2000处的造型的表头，190句为计算字符造型数据实际的地址。210—250循环为将造型数据放入用户建立的造型表中。

400—450子程序为设置造型表表头及每个造型索引项。

460—480为存盘子程序，NU－N表示造型表的长度，存盘名字由用户输入。

以上可进行键盘大写字母及其它字符的编辑。如需要小写字母，则需加下面两条语句并将原110语句删除。

```
110   IF E% > = 33 AND E% < = 64
       THEN J% = E% - 32: GOSUB 15
      0
115   IF E% > = 65 AND E% < = 90
       THEN J% = E% - 6: GOSUB 150
```

第四节　造型表的合并、造型的拼积程序

有时在生成造型表时，常把一些经常使用的子图形生成几个小图库，然后实行积木式拼图。有时也需要将两个造型

表合并为一个造型表或者分解一个造型表为两个造型表。下面就来谈谈这方面的交互程序。

① 程序要求

能重排造型表的次序，将两个或多个造型表合并为一个造型表。

能将两个或多个造型按一定要求合并为一个造型。

有存盘及删除功能。

② 源程序（见GP 1304）

```
5  REM  GP1304
10  DIM B%(50):D$ =  CHR$ (4)
20  SCALE= 1: ROT= 0: HCOLOR= 3
30 K1 = 232:K2 = 233: POKE K1,0
40 N = 24576:M% = 760:N1 = 16384:N%
     = N1
50  INPUT "Enter the numbers of sha
     pe-table:";M
60  POKE N,M:NU = N + 2 + 2 * (M +
     4): GOSUB 160
70  HOME : VTAB 24: INPUT "Command:
     ";A$
80  IF A$ = "MX" THEN  GOSUB 580
90  IF A$ = "DEL" THEN  GOSUB 410
95  IF A$ = "CONT" THEN  GOSUB 160:
     GOSUB 210
100  IF A$ = "AD" THEN  HCR : GOSUB
     210
120  IF A$ = "BLOAD" THEN  COSUB 37
     0
130  IF A$ = "END" THEN  COCUB 340:
     END
140  GOTO 70
150  END
160 NU = NU + 1: POKE NU,0
```

```
165 NU = NU + 1
170 W% = NU - N:J = J + 1: POKE N +
     1,J
180 S% = 0:A =  INT (W% / 256):B =
     W% - A * 256
185  POKE N + 2 * J,B: POKE N + 2 *
     J + 1,A
190  POKE NU + 1,0
200  RETURN
210  HOME : VTAB 24: INPUT "Shape="
     ;A$
220 J =  VAL (A$)
230  IF A$ = "Q" THEN  RETURN
240  IF J < = 0 OR J >  PEEK (N% +
     1) THEN  CALL 65338: GOTO 210
250  POKE K2,64: HCOLOR= 3
255  DRAW J AT 100,100
260 P =  PEEK (49152)
270  IF P = 78 THEN  HGR : GOTO 210
280  IF P = 81 THEN  GOSUB 440: HGR
     : GOSUB 320: RETURN
290  HCOLOR= 0: DRAW J AT 100,100
300  IF P > 128 THEN  POKE 49168,0
310  GOTO 250
320  HGR :J =  PEEK (N + 1)
330  HCOLOR= 3: ROT= 0: POKE K2,96
332  DRAW J AT 100,100
335  RETURN
340  TEXT : INPUT "Enter NAME--";A$

350  PRINT D$;"BSAVE";A$;",A";N;",L
     ";NU - N + 10
360  RETURN
370  HOME : VTAB 24: INPUT "Enter c
     -blocks NAME---";A$
390  PRINT D$;"BLOAD";A$;",A";N1
```

```
400  RETURN
410  IF J2 < 0 THEN  GOTO 430
420 NU = B%(J2 - 1):J2 = J2   1
425  POKE NU + 1,0: HGR : GOSUB 320

430  RETURN
440  IF J2 < 0 THEN  RETURN
450 B%(J2) = NU
460 A1 =  PEEK (N% + 2 * J)
465 B1 =  PEEK (N% + 1 + 2 * J)
467 C1 =  PEEK (N% + 2 + 2 * J)
470 A1 = N% + A1 + B1 * 256
480  IF  PEEK (A1) = 0 THEN A1 = A1
      + 1
490  FOR I = 0 TO 12000
500 A =  PEEK (A1 + I)
510  IF A = 0 THEN 540
520  IF B$ <  > "Y" THEN  POKE NU +
     I,A
530  NEXT I
540 NU = NU + I   1: POKE NU + 1,0
550 J2 = J2 + 1
560  GOSUB 320: RETURN
580 B% = NU
585  IF  PEEK (NU) <  > 0 THEN NU =
     NU + 1
590  GOSUB 320
600  GET B$
610  IF B$ = "I" THEN  POKE NU,128:
     NU = NU + 1: GOTO 710
620  IF B$ = "J" THEN  POKE NU,3:NU
      = NU + 1: GOTO 710
630  IF B$ = "K" THEN  POKE NU,83:N
     U = NU + 1: GOTO 710
640  IF B$ = "L" THEN  POKE NU,1:NU
      = NU + 1: GOTO 710
650  IF B$ = "E" THEN  FOR I = 0 TO
```

```
        5: POKE NU + I,128: NEXT :NU =
        NU + 6: GOTO 710
660     IF B$ = "S" THEN S1 = 219: POKE
        NU,S1: POKE NU + 1,S1NU = NU +
        2: GOTO 710
670     IF B$ = "D" THEN S1 = 146: POKE
        NU,S1: POKE NU + 1,S1:NU = NU +
        2: GOTO 710
680     IF B$ = "F" THEN S1 = 73: POKE
        NU,S1: POKE NU + 1,S1:NU = NU +
        2: GOTO 710
690     IF B$ = "B" THEN  GOSUB 770
700     IF B$ = "Q" THEN  RETURN
710     POKE NU,45: POKE NU + 1,0
720     GOSUB 320
730     GOTO 600
740     FOR I = 0 TO 8:S =  PEEK (786 +
        I): POKE NU + I,S: NEXT
750     POKE NU + 8,0
760     RETURN
770     IF NU <  = B% THEN  CALL 65338
        : RETURN
780     POKE NU,0:NU = NU - 1
790     HCOLOR= 3: DRAW J AT 100,100
800     RETURN
```

③ 程序说明

5—150为主程序，60句为设置造型表表头空间及第一个造型的索引项，按MX回车进入移标模块，按DEL回车进入删除模块，按CONT回车进入重排造型表功能，按AD回车进入拼积模块，按BLOAD回车将子图形库调入＄4000为起始处。按END回车进入存盘模块并结束运行。

160—200为设置造型索引项的子程序。

210—340子程序为选子图块的人机交互界面模块。选中子图块后，在屏幕闪烁地显示被选中的造型，按N键重新选，

按Y键表示选中并进入拼积功能。

320—335为显示被积以后的图形子程序。

340—360为存盘子程序。

370—400子程序将子图库调进＄4000为起始处，该图库由造型表中的各造型组成，可达到8KB的空间。

410—430为删除子程序，可删除被积的造型。

440—560子程序为将图库中的造型的二进制编码数据放进用户的图表中。

580—730为移标子程序，可以准确地交互确定造型矢量的终点位置，以便下个造型的拼积，按I键向上移动1个不画向量，按J键向左移动1个不画向量，按K键向下移动1个不画向量，按L键向右移动1个不画向量，按E键向上移动6个不画向量，按S键向左移动6个不画量，按D键向下移动6个不画向量，按F键向右移动6个不画向量，按B键删除当前的不画向量，按Q键退出。

第四章　图形的组织

前面第二、三章我们给出了用点线法和造型法的有关实用工具。组织一幅图形可通过点线法来交互完成，用设置文件结构来存贮有关节点数据，也可以通过造型法来生成一幅图形，它存贮的是图形轨迹的二进制编码（造型表）。前者生成图形的速度慢，后者具有较大的灵活性，并且显示图形速度快，便于实现复杂图形的动画。本章主要介绍从数据结构角度如何来组织多幅图形，这里主要针对造型法，把每个造型作为子图形来处理，对于用点线法，介绍的方法也完全实用。

第一节　用数据结构描述图形

在进行交互图形设计时，生成的图形数据不作为单个项，而是在一定的结构上加以处理，实际上，我们在前面的图形交互程序中，已经用到了数组、文件，并涉及到其相应的操作。在进行数据操作时，需要一种特有变量，叫指针，指针变量指向计算机内存中另一个变量的位置。由于BASIC语言中不能使用指针，下面介绍的程序都是通过数组的下标及变量来确定指针。

一、图形菜单的数据格式

图1.8是通用造型软件包的主菜单，其汉字为造型汉字，每个汉字作为一个造型，以左上角为造型的起点，共有25个

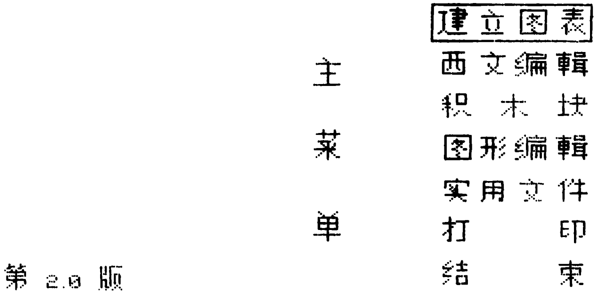


图 1.8 图形汉字菜单屏幕

造型，在造型表中排列顺序为：建、图、表、立、编、辑、积、木、块、打、印、实、用、文、件、形、束、主、单、菜、第、版、2.0、结、□、和西。

显示图1.8的程序（见GP 1401）。

```
5   REM   GP1401
6   DATA  3,255,4,3,2,238,5,3,4,218,
      5,3,1,202,4,3,6,254,27,3,5,236
      ,27,3,14,220,27,3
7   DATA   26,203,28,3,9,254,49,3,7,
      203,49,3,8,229,49,3,6,254,71,3
      ,5,236,71,3,16,221,72,3
8   DATA  2,205,71,3,15,255,94,3,13,
      221,95,3,13,221,95,3,14,237,94
      ,3,12,205,94,3,11,255,116,3
9   DATA 10,203,114,3,17,255,137,3,2
      4,203,137,3,18,147,30,3,20,147
      ,68,3
10   DATA   19,146,111,3,22,51,140,3
      ,23,28,151,3,21,10,139,3,25,19
      9,2,3
20   DIM A%(35),B%(35),C%(35)
30   FOR I = 1 TO 31
```

```
40  READ A%(I),B%(I),C%(I),E: NEXT
45  HGR2 : SCALE= 1
50  HCOLOR= 3: ROT= 0
60  POKE 232,0: POKE 233,96
70  FOR I = 1 TO 31
80  DRAW A%(I) AT B%(I),C%(I)
90  NEXT
100  END
```

程序GP 1401调用各造型的数据放在A%(I)、B%(I)、C%(I)数组中，各造型颜色均为3号颜色。这些数据可以按照下面约定结构来描述。见图1.9。

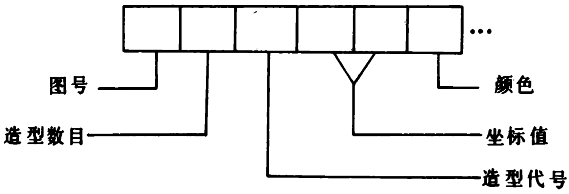


图 1.9　图形数据格式

第一位为图号，第二位为该图号所要显示的造型数，后绪每四位表示调用某个造型的代号、位置坐标和颜色。上述结构可以描述由造型来组织的一幅图形，用一个A%(I)数组来表示该种结构，A%(0)表示图号，A%(1)表示该图形包含的造型数，其显示程序如GP 1402所示。

```
4  REM  GP1402
5  DATA    1,31
6  DATA  3,255,4,3,2,238,5,3,4,218,
       5,3,1,202,4,3,6,254,27,3,5,236
```

```
,27,3,14,220,27,3
7  DATA   26,203,28,3,9,254,49,3,7,
   203,49,3,8,229,49,3,6,254,71,3
   ,5,236,71,3,16,221,72,3
8  DATA  2,205,71,3,15,255,94,3,13,
   221,95,3,13,221,95,3,14,237,94
   ,3,12,205,94,3,11,255,116,3
9  DATA 10,203,114,3,17,255,137,3,2
   4,203,137,3,18,147,30,3,20,147
   ,68,3
10  DATA   19,146,111,3,22,51,140,3
   ,23,28,151,3,21,10,139,3,25 ,1
   99,2,3
20  DIM A%(200)
25  READ A%(0),A%(1)
30  FOR I = 2 TO A%(1) * 4 + 1
40  READ A%(I): NEXT
45  HGR2 : SCALE= 1
50  HCOLOR= 3: ROT= 0
60  POKE 232,0: POKE 233,96
70  FOR I = 1 TO 31
80  DRAW A%(4 * I - 2) AT A%(4 * I -
   1),A%(4 * I)
90  NEXT
100  END
```

运行GP 1401和GP 1402需要上述25个汉字造型的造型表。

二、用二维数组来描述多幅图形

①关系结构

前述图形汉字屏幕是由造型组成的一幅图形，可用一维数组来存贮有关数据，数组的维大小由造型数×4＋1来设定。如果是多幅图形，我们可以用二维数组A％（J,I ）来描述，J 行表示图形个数，I 列存放相应图形的有关数据。

我们知道，数组在内存存贮一片连续单元，对二维数组，因为我们设置维大小时应按每幅图形中能存贮的最大数来确定的。实际情况下，有的图形可能有几个造型数据，这样势必造成内存的浪费，但另一方面，在实现数据的操作时，二维数组描述的图形较为方便，下面介绍该种结构数据操作。

②二维数组存贮图形数据的几种操作

增加，在某个图形数据后面增加相应的图形数据。比如在第一幅图形后面增加8个数据，其执行细节为：

· 由1号确定A%(1，I)，并确定I变量值I_0。

· 由A%(1，I_0)起顺序输入8个数据。

· 将A%(1，I_0+9)置零，同时将A%(1，1)⇐A%(1，1)+8。零表示该数据的结束。

遍历，将二维数组中的每个数据打印在屏幕上，执行一个二重循环，当内循环执行遇到后面数据项为零则退出该循环。

删除，删除某个图形实际上是将该图形数据项全部清零或者将其关键字置零，比如要删除第八幅图形，其执行细节为：

· 由第八个图形确定A%(8，I)。

· 将A%(8，I)全部置零或者将关键字A%(8,1)和A%(8，2)置零。

替换，替换某幅图形中的图形，实际上是将该图形的子图形数据用别的子图形数据代替。比如要替换第三个图形中的第二个子图形，若子图形为造型，则占用4个字节，其执行细节为：

· 由第三个图形确定A(3，I)。

·将A％(3，8)，A％（3，9），A％（3，10）和A％（3，11）分别用另一个造型的4个数据代替。

③源程序（见GP 1403）

```
5   REM   GP1403
10   DIM A%(10,100)
20   VTAB 24: INPUT "command:";A$
25   IF A$ < "1" OR A$ > "4" THEN
       CALL 65338: GOTO 20
30   IF A$ = "1" THEN  GOSUB 400
40   IF A$ = "2" THEN  GOSUB 600
50   IF A$ = "3" THEN  GOSUB 800
55   IF A$ = "4" THEN  GOSUB 900
60   GOTO 20
70   END
100   FOR I = 1 TO 100
110  B = A%(J,I)
120   IF B ≠ 0 THEN I0 = I: GOTO 1
      40
130   NEXT
140   RETURN
400   REM  REPEND
410   VTAB 24: INPUT "JJ=";J
412   IF J < 1 OR J > 10 THEN  CALL
      65338: GOTO 410
420   GOSUB 100
430   INPUT "REPEND NUMBER";NUM
435   IF NUM + I0 > 100 THEN 430
440   FOR I = I0 TO NUM + I0 - 1
450   INPUT A%(J,I)
460   NEXT
470   RETURN
600   REM  DELETE
610   VTAB 24: INPUT "JJ=";J
620   FOR I = 1 TO 100
630   PRINT A%(J,I);"  ";
```

```
640  NEXT
645  PRINT
650  PRINT "Is the numbers?<Y/N>"
660  GET A$
670  IF A$ = "N" THEN  CALL 65338
     : GOTO 610
675  RETURN
680  FOR I = 1 TO 100
690 A%(J,I) = 0
700  NEXT
710  RETURN
800  REM  TRAVERSE
802  FOR J = 1 TO 10
805  PRINT "J=";J;"--";
810  FOR I = 1 TO 100
815 B = A%(J,I)
820  IF B = 0 THEN 840
825  PRINT B;" ";
830  NEXT I
840  PRINT
850  NEXT J
860  RETURN
900  REM   REPLACE
910  VTAB 24: INPUT "JJ=";J
920  INPUT "II=";I
930  PRINT "A%("J","I")=";A%(J,I)

940  PRINT "Is the number?<Y/N>"
945  GET A$
960  IF A$ = "Y" THEN  GOSUB 980
970  IF A$ = "N" THEN  CALL 65338
     : GOTO 910
975  RETURN
980  INPUT A%(J,I)
990  RETURN
```

④程序说明

5—70为主程序，用以对二维数组A％（J，I）施行下述操作，当A＄＝“1”时，则执行增加操作；当A＄＝“2”时，则执行删除操作；当A＄＝“3”时，则执行遍历操作；当A＄＝“4”时，则执行替换操作。

400—470为增加子程序，其中100—140子程序为找出变量I_0，数据以零为结束标志。

600—710为删除子程序，选中要删除对象时，将其全部置零。

800—860为遍历子程序，将所有的数据显示在屏幕上。

900—990为替换子程序，用以替换指定的A％（J,I）之中的某一个数据。

三、用索引结构来描述多幅图形

①变形树结构

对上述的用二维数组来描述图形的结构，我们也可以用一维数组来描述，并设置一个索引表头存放各图形指针，A％(I)数组存放各图形指针，B％(I)数组存放各图形数据（见图1.10）。

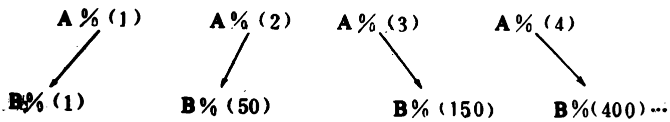


图 1.10 两个数组中数据之间的对应关系

这种结构类似于变形树图形数据结构，其结构图见图1.11，这种结构充分利用内存，各图形具有不等长的结构，每个图形数据大小由子图形个数多少来确定，这种结构适用

于内存小的图形结构描述，下面看看该种结构的几种操作。

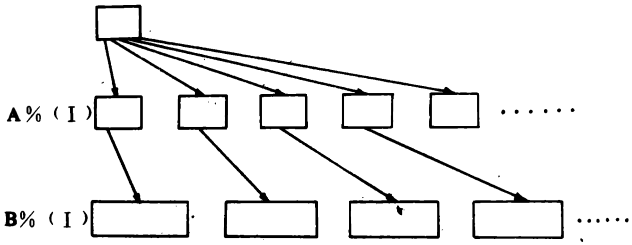


图 1.11 树形结构

②索引变形树结构存贮图形数据的几种操作

增加，在某个图形数据后面增加相应的图形数据，对已经建立好的图形，由于数据实行紧凑格式排列，若要增加，一方面要改变后绪图形的指针；另一方面对后续图形数据要向后搬移。若对新建立的图形增加相应的子图形数据，则不存在上述问题。其操作类似于二维数组描述的图形数据的增加操作。

遍历，将存放图形数据按一定格式显示在屏幕上，执行二重循环、其内循环的循环起始变量及终止变量即为该图形及相应图形指针变量。

删除，删除某个图形实际上改变该图形的指针变量或者通过指针变量将其图形数据全部置零，比如要删除第二个图形中的第八个造型，其执行细节为：

· 由A%(2)找到第二个图形数据的起始位置B%(50)。

· 由B%(50)确定被删除造型的位置，即B%(50+4×8+2)。

由B%（50+4×8+2）处删除相应的造型数据，

并将后续图形数据分别向前移动 4 位。

· 造型数减1，即B％（51）＝B％（51）－1的操作。

替换，用别的子图形数据代替某幅图形中的子图形数据。比如我们要替换第一个图形中的第八个造型。其执行细节为：

·由A％(1)找到第一个图形数据的起始位置B％(1)。

·由B％(1)确定被替换造型的位置，即B％(1＋4×8＋2)。

·由B％(1＋4×8＋2)处，用其它造型数据替换该处的造型。

我们还可以对这种结构进一步改进，比如在索引项里安排了一些插入指针，可以连续插入多个数据，在内存允许情况下，我们还可以对图形数据采用链结构，即每个造型数据安排一个指针，这样删除、增加操作都比较方便。

③源程序（见GP 1404）

```
5  REM   GP1404
10  DIM A%(11),B%(500)
20  VTAB 24: INPUT "Command:";A$
25  IF A$ < "1" OR A$ > "4" THEN
      CALL 65338: GOTO 20
30  IF A$ = "1" THEN  GOSUB 400
40  IF A$ = "2" THEN  GOSUB 600
50  IF A$ = "3" THEN  GOSUB 800
55  IF A$ = "4" THEN  GOSUB 900
60  GOTO 20
400  REM  REPEND
410  VTAB 24: INPUT "JJ=";J
412  IF J < 1 OR J > 10 THEN  CALL
     65338: GOTO 410
420 I0 = A%(J + 1)
430  INPUT "REPEND NUMBER";NUM
```

```
435  IF NUM + I0 > 500 THEN 430
440  FOR I = I0 TO NUM + I0 - 1
450  INPUT B%(I)
460  NEXT
470 A%(J + 1) = I0 + NUM
480  RETURN
600  REM  DELETE
610  VTAB 24: INPUT "JJ=";J
615 I0 = A%(J):I1 = A%(J + 1)
620  FOR I = I0 TO I1
630  PRINT B%(I);" ";
640  NEXT
645  PRINT
650  PRINT "Is the numbers?<Y/N>"
660  GET A$
670  IF A$ = "N" THEN  CALL 65338
     : GOTC 610
680  FOR I = I0 TO I1
690 B%(I) = 0
700  NEXT
710  RETURN
800  REM  TRAVRSE
802  FOR J = 1 TO 10
803  PRINT "J=";J;"....";
805 I0 = A%(J):I1 = A%(J + 1)
807  IF I1 = 0 THEN 840
810  FOR I = I0 TO I1
820  PRINT B%(I);" ";
830  NEXT I
840  PRINT
850  NEXT J
860  RETURN
900  REM   REPLACE
910  VTAB 24: INPUT "JJ=";J
915 I0 = A%(J)
920  INPUT "II=";I
930  PRINT "B%("I0 + I;")";B%(I0 +
```

```
       I)
940    PRINT "Is the number?<Y/N>"
945    GET A$
960    IF A$ = "Y" THEN  GOSUB 980
970    IF A$ = "N" THEN  CALL 65338
       : GOTO 910
975    RETURN
980    INPUT B%(I0 + I)
990    RETURN
```

④程序说明

5—70为主程序，A%(I)存放各图形指针，B%(I)存放图形数据，可实现下述操作：当A$ ＝“1”时，则执行增加操作；当A$ ＝“2”时，则执行删除操作；当A$ ＝“3”时，则执行遍历操作；当A$ ＝“4”时，则执行替换操作。

400—480为增加子程序，每个图形数据以零为结束标志。

600—710为删除子程序，当选中要删除对象时，将其全部置零。

800—860为遍历子程序，将所有数据按变长结构显示在屏幕上。

900—990为替换子程序，用以替换指定的B%(I)之中的某一个数据。

第二节　数据的存贮方式

前节谈到了实现图形数据的几种结构及其相应的操作。对二维数组可以用随机文件来存贮，随机文件的记录长度即为维数大小，每个记录对应一幅图形。对索引变形树结构的数据，用顺序文件来存贮，顺序存贮索引指针表头及相应的

图形数据数组。也可以在内存开设缓冲区来存放图形数据，内存缓冲区的位置可以灵活地放在内存的可用区域，也可以直接放在造型表之后，并能同造型表一起存盘，下面主要针对后一种方法看看存贮数据的实现。

一、将变量或数组中的数据放入内存

前面介绍程序是用数组来存贮图形数据的，我们设想将数组的连续存贮区直接送入和造型表相近的一段区域，不就能够同造型表一起存盘了吗？这里我们用另外一种方法，直接将变量或数组用POKE语句放入内存。由于中华学习机是八位机，一个字节可以存放最大整数为255，对于大于255的整型数需要二个字节来存放，有关调用造型涉及到的都是整型数，下面主要谈整型变量及整型数组存入内存的方法。

①将变量放入内存

对于小于等于255的整数，放入指定的内存，若将45放入地址为24567处，为：

```
] A =45
] POKE  24567 , A
```

取出这个数的操作为：

```
] A = PEEK (24567)
] PRINT  A
45
```

对于大于255的整型数，需拆成两个整型数来存放，若将2567存入指定内存的操作为：

```
IPOKE  24576,2567-INT (2567/256) * 256
IPOKE  24577,INT (2567/256)
```

取出这个数的操作为：

IA=256*PEEK(24577)+PEEK(24576)

IPRINT A

2567

② 将数组放入内存

前面介绍的变形树图形数据结构，A%(I)存放索引表头，B%(I)存放图形数据，若A%(0)存该数组的长度，B%(0)也存放该数组的长度，若将这两个数组中的数据存入指定地址为起始处的内存，相应程序见GP1405。

```
5   REM   GP1405
7   DIM A%(200),B%(200)
10   FOR I = 1 TO A(0)
20   POKE NU + 2 * I,A%(I) -  INT
      (A%(I) / 256) * 256
30   POKE NU + 2 * I + 1, INT (A%(
      I) / 256)
40 NU = NU + 2
50   NEXT
60   FOR I = 1 TO B%(0)
70   POKE NU + 2 * I,B%(I) -  INT
      (B%(I) / 256) * 256
80   POKE NU + 2 * I + 1, INT (B%(
      I) / 256)
90 NU = NU + 2
100   NEXT
110   END
```

说明：NU为开设缓冲区的起始地址。

10—50为存放A%(I)数组中的数据，每存一次，变量NU自动加2。

60—100为存放B%(I)数组中的数据，存完以后变量NU指向缓冲区的尾部。

二、将数据放入造型表之后

设造型表放在＄6000为起始处，将数据放入造型表以后，关键是找出造型表结束的地址，可以进入监控状态后，用查 11）阅地址的方法来算出，而程序ＧＰ 1406可以自动算出造型表结束的地址。

① 源程序

```
5   REM    GP1406
10 N = 24576
20  INPUT "Enter NAME:";A$
30  PRINT  CHR$ (4);"BLOAD";A$;"A
     $6000"
50 J =  PEEK (N + 1)
60 A3 =  PEEK (N + 2 * J)
65 B3 =  PEEK (N + 2 * J + 1)
70 NU = N + A3 + B3 * 256
80  FOR I = 0 TO 13000
90 A2 =  PEEK (NU + I)
100  IF A2 = 0 THEN 120
110  NEXT I
120  PRINT "NU=";NU
130  END
```

② 程序说明

运行该程序，让用户输入造型表的名字，该造型表放在＄6000为起始处，造型表的造型数必须放在＄6001地址。

5—7为计算造型表中最后一个造型的起始地址。

8—12为计算最后一个造型的结束地址。

注意：运行该程序，实际造型数目必须存入造型表开始的第二个字节内。

第三节　图形数据的交互生成程序

上面介绍了几种图形数据结构及相应存贮方式，但如何来生成相应的数据结构，最笨的办法是通过向数组里输入有

关数据，但这样不比编调用造型的程序来得方便，下面介绍如何对图形的动态操作来交互生成相应的图形数据结构。并给出相应的程序。

一、程序结构及主要功能

① 程序结构

本程序采用树型层次结构的菜单方式，其模块结构见图1.12。

② 程序功能

· 对已建立造型表中的造型实现动态的复制、增加、删除等操作。

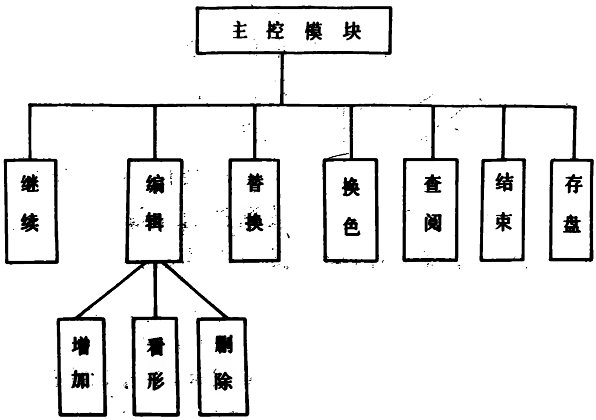


图 1.12　图形交互生成工具模块图

· 具有查阅、替换、幻灯片功能。

· 可建立多幅图形，图形个数主要取决于内存大小。

· 能将图形数据与造型表一起存盘。

二、生成图形的数据结构

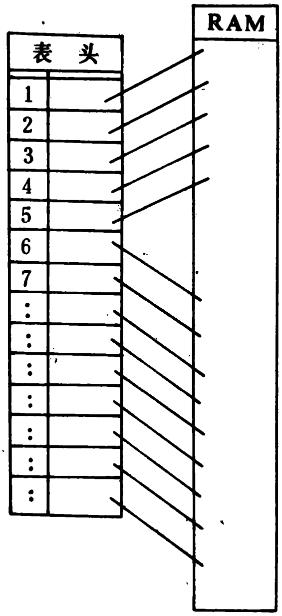


图 1.13 图形数据结构

生成的图形的结构采用索引文件的变形树结构，为了程序简便起见，各图形中的子图形没有安排指针。相应的就没有插入操作。该种结构直接放在造型表之后，这种结构的操作类似于一般高级语言所提供的随机文件，但其操作均在RAM内进行，且各记录可按变长存贮，理论上各记录的长度不是事先规定，而是随着图形的增加自动增加的，一个图块的复制占4个字节，其结构见图1.13。

本结构存取迅速，各图形数据具有不等长，且连续存贮在造型表之后，对较小内存的微机尤为实用。

① 源程序（见GP 1407）

```
5  REM   GP1407
10  DATA   1,0,4,0,36,54,45,63,63
     ,45,54,4,0
12  FOR I = 0 TO 12
15  READ A: POKE 768 + I,A: NEXT

18 D$ =  CHR$ (4)
20  SCALE= 1: ROT= 0: HCOLOR= 3
22 HC = 3:N = 24576: GOSUB 1300
30 K1 = 232:K2 = 233: POKE K1,0
55  POKE N1 - 1,0: POKE N1 - 2,0:
```

```
     POKE N1 - 3,0: GOSUB 1515
60  GOSUB 170
65  VTAB 22: INPUT "Enter slectio
     n:";A$
67  IF A$ = "1" THEN  GOSUB 1500
75  IF A$ = "2" THEN A = 2: GOSUB
     700: GOSUB 1100
80  IF A$ = "3" THEN A% = 1:A = 3
     : GOSUB 700: HGR2 : GOSUB 85
     0:A = 2: GOSUB 700: GOSUB 11
     00
82  IF A$ = "4" THEN  GOSUB 600
85  IF A$ = "5" THEN  GOSUB 1400
90  IF A$ = "6" THEN  GOSUB 1600
95  IF A$ = "7" THEN  HOME : END

120  GOTO 65
125  END
170  HOME : TEXT : VTAB 4
178  PRINT "1. Continue"
179  PRINT "2. Great a drawinq "
180  PRINT "3. Repleas a shape in
      a drawinq"
185  PRINT "4. Change the drawinq
     -color"
190  PRINT "5. Look at the drawin
     q-table"
195  PRINT "6. Save the drawing-t
     able to disk"
200  PRINT "7. End"
600  TEXT : HOME : VTAB 22: INPUT
     "Color<0,1,2,3,4,5,6,7>--";H
     C
605  IF HC < 0 OR HC > 7 THEN  CALL
     65338: GOTO 600
610 I2 = 2: GOSUB 170: RETURN
700  HGR : HOME : VTAB 23
```

```
705  IF A = 2 THEN  INPUT "Shape=
     ";A$
710  IF A = 3 THEN  INPUT "drawin
     g=";A$
715 J =  VAL (A$)
720  IF A = 2 THEN  IF J <  = 0 OR
     J >  PEEK (N) THEN  CALL 653
     38: GOTO 700
722  IF A = 3 THEN  IF J <  = 0 OR
     J >  PEEK (C0 - 3) THEN  CALL
     65338: GOTO 700
740  IF A = 2 THEN  HCOLOR= 3: POKE
     K2,96: DRAW J AT 100,100
745  IF A = 3 AND A% = 1 THEN J4 =
     J: GOSUB 1400
750 P =  PEEK (49152)
760  IF P = 78 AND A = 2 THEN 700

765  IF P = 78 AND A = 3 THEN 700

780  IF P = 81 THEN  HGR2 : RETURN

790  HCOLOR= 0: DRAW J AT 100,100

800  IF P > 128 THEN  POKE 49168,
     0
810  GOTO 740
850 K0 =  PEEK (N1 - 2 * J - 4) +
     256 *  PEEK (N1 - 5 - 2 * J)
     :J% = K0
860  POKE K2,96
870  FOR I = K0 TO 1000
880 A2 =  PEEK (N1 + 4 * I):B2 =
      PEEK (N1 + 1 + 4 * I):C2 =
      PEEK (N1 + 2 + 4 * I)
890  HCOLOR=  PEEK (N1 + 3 + 4 *
     I)
```

```
900  IF A2 = 0 AND B2 = 0 THEN L%
      = I + 10:K% = I: RETURN
910  IF A% < > 1 THEN 940
920  GET A$
930  IF A$ = "Q" THEN B = 3:K% =
     I - 1: RETURN
940  DRAW A2 AT B2,C2
950  NEXT I
960  RETURN
970  POKE K2,96
980 K =  PEEK (N1 - 2 * J4 - 4) +
     256 *  PEEK (N1 - 5 - 2 * J4
     )
990  HCOLOR= 3: SCALE= 1
1000  FOR I = K TO 2000
1010 A2 =  PEEK (N1 + 4 * I):B2 =
      PEEK (N1 + 1 + 4 * I):C2 =
      PEEK (N1 + 2 + 4 * I)
1020  HCOLOR=  PEEK (N1 + 4 * I +
     3)
1030  IF A2 = 0 AND B2 = 0 THEN 1
     050
1040  DRAW A2 AT B2,C2: NEXT I
1050  RETURN
1060  IF X >  = 255 THEN X = 255
1070  IF X < 0 THEN X = 0
1080  IF Y > 191 THEN Y = 190
1085  IF Y < 0 THEN Y = 0
1090  RETURN
1100 C1 = 10:X   100:Y = 100
1120  IF A% = 1 THEN J0   K%
1130  GET A$
1140  IF J <  =  PEEK (N + 1) THEN
      POKE K2,96: HCOLOR= 0: DRAW
     J AT X,Y: HPLOT X,Y
1150  IF A$ = "I" THEN Y = Y - C1
     : GOTO 1260
```

```
1160  IF A$ = "J" THEN X = X - C1
     : GOTO 1260
1170  IF A$ = "L" THEN X = X + C1
     : GOTO 1260
1180  IF A$ = "M" THEN Y = Y + C1
     : GOTO 1260
1190  IF A$ =  CHR$ (8) AND A = 2
      AND B <  > 3 THEN  GOSUB 16
     60
1200  IF A$ = "F" THEN C1 = C1 +
     1
1210  IF A$ = "" THEN C1 = 1
1220  IF A$ = "A" AND A% <  > 1 THEN
     J4 =  PEEK (N1 - 3): GOSUB 9
     70
1230  IF A$ = "A" AND A% = 1 THEN
      GOSUB 970
1240  IF A$ = "E" AND A = 2 AND B
      <  > 3 THEN  HCOLOR= 3: POKE
     K2,3: DRAW 1 AT X,Y: GOSUB 1
     710
1250  IF A$ = "Q" THEN A = 0:A% =
     0: GOSUB 170:J0 =  PEEK (N1 -
     1) + 256 *  PEEK (N1  2):B =
     0:HC = 3: RETURN
1260  GOSUB 1060
1270  IF A = 2 AND B <  > 3 THEN
     1290
1280  POKE N1 + 4 * J0,J: POKE N1
      + 1 + 4 * J0,X: POKE N1 + 2
      + 4 * J0,Y: POKE N1 + 3 + 4
      * J0 + 1,HC
1290  IF J <  =  PEEK (N + 1) THEN
      HCOLOR= 3: POKE K2,96: DRAW
     J AT X,Y: HPLOT X,Y
1295  GOTO 1130
1300 J =  PEEK (N + 1)
```

```
1320 A3 =  PEEK (N + 2 * J):B3 =
      PEEK (N + 2 * J + 1)
1330 NU = N + A3 + B3 * 256
1340  IF  PEEK (NU) = 0 THEN NU =
     NU + 1
1350  FOR I = 1 TO 13000
1360 A2 =  PEEK (NU + I)
1370  IF A2 = 0 THEN 1390
1380  NEXT I
1390 N1 = I + NU + 600
1395  RETURN
1400  POKE K2,96:J =  PEEK (N1 -
     3)
1420  FOR J4 = 1 TO J
1430  HGR2
1440  HCOLOR= 3: GOSUB 970
1450  GET B$: IF B$ = "Q" THEN 14
     70
1460  NEXT J4
1470  SCALE= 1: GOSUB 170: RETURN
1500  PRINT "Continue?<N>": GET A
     $
1505  IF A$ = "N" THEN  GOSUB 170
     : RETURN
1510  HOME : GOSUB 170:J1 = 0
1515  IF J0 > 1 THEN J0 = J0 + 2
1520  IF J0 = 0 THEN J0 = J0 + 1
1530 J3 =  PEEK (N1 - 3):J3 = J3 +
     1
1540  FOR I = 0 TO 12: POKE N1 +
     4 * J0 + I,0: NEXT
1550 A1 = J0 -  INT (J0 / 256) *
     256:B1 =  INT (J0 / 256)
1560  POKE N1 - 4 - J3 * 2,A1
1570  POKE N1 - 1,A1: POKE N1 - 2
     ,B1
1580  POKE N1 - 3,J3:J5 = J0
```

```
1590  RETURN
1600  HOME : VTAB 24: TEXT
1610  INPUT "Enter NAME ------:";A
     $
1620 A$ = A$ + ".TAB"
1630  PRINT D$;"BSAVE";A$;",A";N;
     ",L";N1 - N + 4 * J0 + 10
1640  GOSUB 170
1650  RETURN
1660  IF J0 < = J5 THEN  CALL 65
     338: RETURN
1670 J0 = J0 - 1:J1 = J1 - 1
1680  HCOLOR= 0: POKE K2,96: DRAW
      PEEK (N1 + J0 * 4) AT  PEEK
     (N1 + 1 + 4 * J0), PEEK (N1 +
     2 + 4 * J0)
1690  GOSUB 1730
1700  RETURN
1710  POKE N1 + 4 * J0,J: POKE N1
      + 1 + 4 * J0,X: POKE N1 + 2
      + 4 * J0,Y: POKE N1 + 3 + 4
      * J0,HC
1720 J0 = J0 + 1
1730  FOR I = 0 TO 12: POKE N1 +
     4 * (J0 + 1) + I,0: NEXT
1740  POKE N1 - 1,J0 -  INT (J0 /
     256) * 256: POKE N1 - 2, INT
     (J0 / 256)
1750  RETURN
```

② 程序说明

5—125为主程序，10—15将十字光标造型数据存入$300为起始处，其中22句为计算出造型表结束地址，具体由子程序1300—1395来完成，55句为对内存初始化并置第一个图形的指针。当A$ = "1",执行继续模块，即设置下个图形的指针，并从下个图形开始；当A$ = "2"时，则执行编

辑功能，当A＄＝“3”时，则执行替换模块；当A＄＝“4”时，则执行换色模块；当A＄＝“5”时，则执行查阅模块；当A＄＝“6”时，则执行存盘模块。

170—200子程序为命令提示项。

600—610为换色子程序，用以改变当前复制造型的颜色，利用此项功能，可以对图形实现局部修改，实际上是抹黑处理。

700—810子程序为交互选定图形或者造型的人机界面程序。被选中的造型将以闪烁的形式显示在屏幕上，当按“Q”键表示选中并退出该子程序，按“N”键重新输入图形或者造型。

850—960子程序为顺序显示一幅图形中的各造型模块，主要用于替换造型的人机界面。当按“Q”键，选中该造型，并记下该造型在图形数据结构中的位置。

970—1050子程序为一幅图形的遍历程序，即显示一幅图形的子程序。

1060—1090子程序为控制光标不要超出屏幕所规定范围程序。

1100—1295子程序为增加、替换、复制、删除程序。按I键被选中造型上移，按J键被选中的造型左移，按M键被选中的造型下移，按L键被选中的造型右移。当按CTRL-H键或←键进入删除操作，在具有替换状态下该命令失效。按CTRL-F键后，使得按一次移动键只移动一格，按F键可增加移动速度。按A键看被操作的图形。按E键在当前图形层上复制被选中的造型，在具有替换状态下该命令失效。按Q键退出该子程序。

1300—1395子程序为计算造型表结束地址，并将该地址加600，一个图形需要两个字节来存放指针，则这600个字节可设置300幅图形指针数据。

1400—1470子程序为查阅模块，按Q键退出该模块，按其它键，顺序显示所建立的图形。

1470—1590子程序为设置每幅图形指针的程序。

1600—1650子程序为存盘模块，将造型表及图形数据一块存盘，并后缀·TAB。

1660—1700子程序为删除模块，由后向前删除复制的造型。

1710—1750子程序为复制模块，将图形数据存入内存。

运行上面程序前，必须将造型表调入＄6000为起始处，且＄6001地址存入造型表实际造型个数。

第四节　几个应用实例

下面介绍借用上述工具程序做具体几个图形应用的实例，这些是复杂的数学公式，图形菜单，复杂图形等。

一、复杂的数学公式

在计算机辅助教学中，常常要建立带不规则符号的函数式子或公式的图形。建立这些不规则的图形有多种方法，用HPLOT语句，或者用一个造型，虽然用拼积功能可以生成一个造型，但要占用较多内存。下面看看如何用工具软件来生成相应的图形数据结构来实现，十分方便也十分可行。

复杂的数学公式一般由多个字符或符号组成的，如果把复杂的数学式子作为图形看待，则每个字符或符号可以作为造型来处理，有关这些造型的生成可用第二部分的诸多工具

程序来实现。然后再用图形数据交互生成程序GP1407对造型表实行编辑，最终生成复杂函数公式的图形。

实例，求三个均质物体截面重心坐标x_c，y_c的公式为：

$$x_c=\frac{x_1s_1+x_2s_2+x_3s_3}{s_1+s_2+s_3}$$

$$y_c=\frac{y_1s_1+y_2s_2+y_3s_3}{s_1+s_2+s_3}$$

其中x_i，y_i为各均质物体形心的坐标，s_i为均质物体面积，试在图形状态下画出上述求截面形心的公式。

解：

1．划分子造型

上述公式的子造型包括：

x，y，c，s，1，2，3，=，＋和——。一共有10个造型。

2．作出子造型

由上面划分的子造型需建立包括10个造型的造型表，借用键盘字符交互生成程序GP1303生成上述10个造型，并按上述排列给予编号。

3．图形生成

用图形数据交互生成程序中的复制模块完成上述公式的生成，见图1.14。

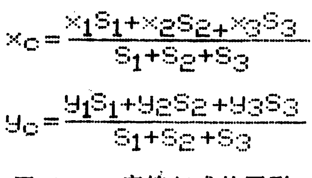


图 1.14 字符组成的图形

4. 将函数式子编在高级语言程序中

将调用图1.13的各造型数据放在数组中，其程序如GP 1408所示。

① 源程序

```
4   REM   GP1408
5   DATA  1,90,80,3,96,82,8,103,80
       ,10,111,78,1,110,73,5,116,76
       ,4,120,73,5,126,76
10   DATA  9,129,73,1,136,73,6,142
       ,76,4,148,74,6,154,76,9,159,
       76,1,166,73,7,171,76
15   DATA  4,177,73,7,183,76,4,127
       ,87,5,133,90,9,136,86,4,143,
       87,6,149,90,9,154,86,4,161,8
       7
20   DATA  7,167,90,2,90,115,3,96,
       116,8,103,113,10,110,111,2,1
       10,107,5,116,109,4,119,106,5
       ,125,109,9,128,106
25   DATA  2,135,107,6,141,108,4,1
       47,106,6,141,108,4,147,106,6
       ,153,108,9,159,106,2,166,107
       ,7,172,108,4,178,106,7,184,1
       08
30   DATA  4,127,120,5,133,122,9,1
       36,119,4,143,120,6,149,122,9
       ,155,119,4,162,120,7,168,122

35   DIM A%(60),B%(60),C%(60)
40   FOR I = 1 TO 54
50   READ A%(I),B%(I),C%(I)
60   NEXT
70   HGR2 : HCOLOR= 3: ROT= 0
80   POKE 232,0: POKE 233,96
90   FOR I = 1 TO 54
```

```
100  DRAW A%(I) AT B%(I),C%(I)
110  NEXT
120  END
```

② 程序说明

5—30语句为调用图1.14各造型数据，调用每个造型需要三个数据，造型代号以其位置坐标。有关各造型的颜色统一用颜色语句来设定。

40—60语句将DATA中的数据放入A%(I),B%(I),C%(I)中。

90—110为显示图1.14的语句。

运行上面的程序需要上述10个造型的造型表。

还可以直接应用内存中的图形数据来显示。

用图形数据交互生成程序GP1407编辑的图形，其数据结构已经放在造型表之后，可以直接用PEEK函数取这些图形数据，其程序见GP1409。

① 源程序

```
5  REM   GP1409
10  HGR2 : SCALE= 1: ROT= 0: HCOLOR=
     3
20  POKE 232,0: POKE 233,96
30 NU = 26852
100  FOR I = 1 TO 1000
110 J =  PEEK (NU + 4 * I):X =  PEEK
     (NU + 4 * I + 1):Y =  PEEK (NU
      + 4 * I + 2):C =  PEEK (NU +
     4 * I + 3)
120  DRAW J AT X,Y
```

```
130  IF J = 0 AND X = 0 THEN 150
140  NEXT I
150  END
```

② 程序说明

30语句NU变量的值由程序GP1406求出，也可以进入监控状态查阅内存后计算出。

100—140为显示图1.14的程序。其变量J为造型代号，X和Y为调用造型的位置坐标。该图形数据以两个零为结束标志。C变量为每个造型的颜色值。如将图1.14从一个位置移动到另一个位置，可改动120语句。比如向右横向移动50，向下纵向移动20。则：

```
120  DRAW  J  AT  X+50, Y+20
```

若要在原位置上逆时针旋转90度，则增加语句。

```
105 ROT=48
145 ROT=0
```

105句将各字符逆时针旋转90度,145句在旋转好之后恢复正常显示。

二、图形汉字屏幕的实现

中华学习机具有汉字功能，但执行由汉字组成的软件速度稍慢，由于汉字是使用文本状态显示，在图形状态下不便于标注汉字。有些情况下，即要显示图形，又要在任意位置上标注汉字，则只能采用图形汉字，用造型来组成汉字，速度能够满足要求，而且造型汉字表可以放在内存的任一区域。有关造型汉字的生成可借用前面的造型生成工具。也能由STC2.0软汉字直接由程序转换成造型汉字，组成相应的汉字屏幕可通过图形数据交互生成工具实现。

1. 由软汉字直接转换成造型的程序

① 源程序（见GP1410）

```
5   REM    GP1410
7   INPUT N
10  DIM A(16,15),B(8):S = 16384
15  POKE S,N: POKE S + 1,N:S = S +
     2
20  FOR I = 1 TO N:A = 2 + N * 2 +
     (I - 1) * 128:B = INT (A /
     256): POKE S,A - B * 256: POKE
     S + 1,B:S = S + 2: NEXT
25  FOR I = 1 TO N:X = 32 * (I -
     1) + 1 + 28672
30  FOR J = 1 TO 30:Y = PEEK (X +
     J)
35  FOR K = 7 TO 0 STEP - 1:B(8 -
     K) = Y > = 2 ^ K:Y = Y - 2 ^
     K * (B(8 - K)): NEXT
37 P = J - 15 * (J > 15):Q = (J <
     = 15) + 9 * (J > 15)
40  FOR K = 0 TO 7:A(Q + K,P) = B
     (K + 1): NEXT
45  NEXT
50  FOR K = 1 TO 7: POKE S,24:S =
     S + 1: NEXT
55  FOR W = 1 TO 16: FOR J = 1 TO
     15: PRINT A(WW,J);: NEXT
57  PRINT : NEXT
60 K = 1
65  FOR M = 1 TO 16
67 P = M / 2 < > INT (M / 2)
70  FOR J = P + 15 * NOT (P) TO
     15 * P + NOT (P) STEP P - NOT
     (P)
75 K = K + 1: IF K > 2 THEN  POKE
```

```
      S,B(2) * 8 + B(1):S = S + 1:
      K = 1
80 B(K) - P * (J <  > 1 AND  NOT
      (P) OR J < 15 AND P) + 2 * (
      J = 15) * P + 2 * (J = 1) *
       NOT (P) + 3 *  NOT (P) * (J
       <  > 1 AND  NOT (P) OR J <
       > 15 AND P) + 4 * A(M,J)
85  NEXT : NEXT
90  POKE S,0:S = S + 1
95  NEXT
100  HGR : HCOLOR= 3: ROT= 0: SCALE=
      1: POKE 232,0: POKE 233,64: FOR
      I = 1 TO N: DRAW WI AT 140,8
      0: GET A$: XDRAW I AT 140,80
      : NEXT
105  TEXT : PRINT : INPUT "The NA
      ME of the shape--";A$: IF A
      $ = "" THEN 115
110  PRINT  CHR$ (4);"BSAVE";A$;"
      ,A$4000";",L";S - 16384 + 5
115  END
```

② 程序说明

使用前，首先把你所想转换成向量的汉字在STC2.0系统下产生一个小字库并存盘，然后返回到DOS系统，把小字库调入内存，运行此程序只要输入小字库中汉字的个数即可。

5—15句为产生向量汉字造型表表头。

20—45句为产生汉字数组表。

50—95为写入造型表内容。

100语句为逐一显示汉字向量。

S变量为造型表的起始地址，数组A(I)为汉字数组表，数

组B(I)为每次读得内存中小字库的内容。

2. 图形汉字屏幕的实现

图1.15是通用造型软件包中建立图表模块的主菜单，建立这个汉字屏幕可按以下步骤来实现。

① 每个汉字用一个造型，各个造型作为子图形，造型表按顺序排列为：继、续、矢、量、直、线、造、圆、弧、椭、辅、助、退、出和 □ ，一共16个造型。

② 按上述排列顺序，由造型生成工具或者由软汉字转换成造型汉字程序生成造型表。

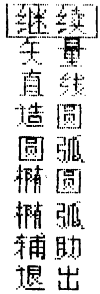


图 1.15 图形汉字屏幕

③ 用图形数据交互生成程序生成图1.15，该汉字菜单在屏幕的右上角。

④ 将这个图形汉字屏幕编在主控程序中，即显示这个汉字屏幕的模块放在主控程序中。

3. 图形汉字区域点菜单的实现

图1.15为汉字菜单，通过屏幕的矩形方框，如何来选中菜单上的某一模块，一方面要控制矩形方框的上移或者下移位置，另一方面在移动过程中要记住汉字菜单与方框重合的位置，区域命令点菜单方式要用位置及键的复合条件来确定。

① 源程序见（GP1411）

```
5  REM   GP1411
6  HGR2
10 CD = 31733:DD = CD + 1:ED = CD
      + 2
15  HCOLOR= 3: SCALE= 1: ROT= 0
```

```
20 K1 = 232:K2 = 233: POKE K1,0
22 I0 = 1: GOSUB 200
25  GET A$
30  IF A$ =  CHR$ (13) AND K = 1 THEN
      GOSUB 500
40  IF A$ =  CHR$ (13) AND K = 2 THEN
      GOSUB 1000
50  IF A$ =  CHR$ (13) AND K = 3 THEN
      GOSUB 1500
60  IF A$ =  CHR$ (13) AND K = 4 THEN
      GOSUB 2000
70  IF A$ =  CHR$ (13) AND K = 5 THEN
      GOSUB 2500
80  IF A$ =  CHR$ (13) AND K = 6 THEN
      GOSUB 3000
90  IF A$ =  CHR$ (13) AND K = 7 THEN
      GOSUB 3500
110  IF A$ =  CHR$ (13) AND K = 9
      THEN  TEXT : END
150  IF A$ = "I" THEN  HCOLOR= 0:
      DRAW 20 AT 232,AD:AD = AD -
     17:K = K - 1: GOSUB 190
160  IF A$ = "M" OR A$ =  CHR$ (3
     2) THEN  HCOLOR= 0: DRAW 20 AT
     232,AD:AD = AD + 17:K = K +
     1: GOSUB 190
170  GOTO 25
180  END
190  IF AD >  = 140 OR AD <  = 0 THEN
     AD = 0:K = 1
191  POKE K2,112
192  HCOLOR= 3: DRAW 52 AT 232,AD

195  RETURN
203  POKE K2,112: FOR I = I0 TO 1
     000
204 A1 =  PEEK (CD + I * 3):B1 =
```

```
     PEEK (DD + 3 * I):C1 =  PEEK
     (ED + 3 * I)
206  IF A1 = 0 AND B1 = 0 THEN AD
      = 0:K = 1: RETURN
208  DRAW A1 AT B1,C1
209  NEXT
```

②程序说明

5—180 为主程序，C D变量为汉字造型表图形数据的起始地址，150 句为使方框上移，160 句为使方框下移。在移动过程中，记下了方框的位置，为了比较方便，设置K变量与方框移动位置一一对应，当K = 1时，对应继续模块；当K = 2时，对应矢量模块；当K = 3时，对应直线模块；当K = 4时，对应椭圆模块；当K = 5时，对应圆弧模块；当K = 6时，对应椭圆模块；当K = 7时，对应椭弧模块；当K = 8时，对应退出模块。

190—195子程序为控制矩形方框不超出汉字菜单之外。

203 —209 子程序为显示这个图形汉字菜单程序。

三、多幅图形的设计

在进行幻灯片演示，或者顺序演示一系列图形来定性表达某一概念时，其中每一幅图形可以用一个造型，但这样即浪费内存，作图效率又低。多幅图形的设计首先是子图形(造型）的划分，然后是图形数据的交互建立，下面用一个实例来谈谈这个

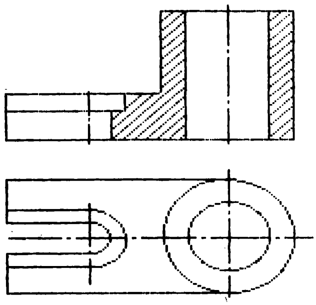


图 1.16　零件的二视图

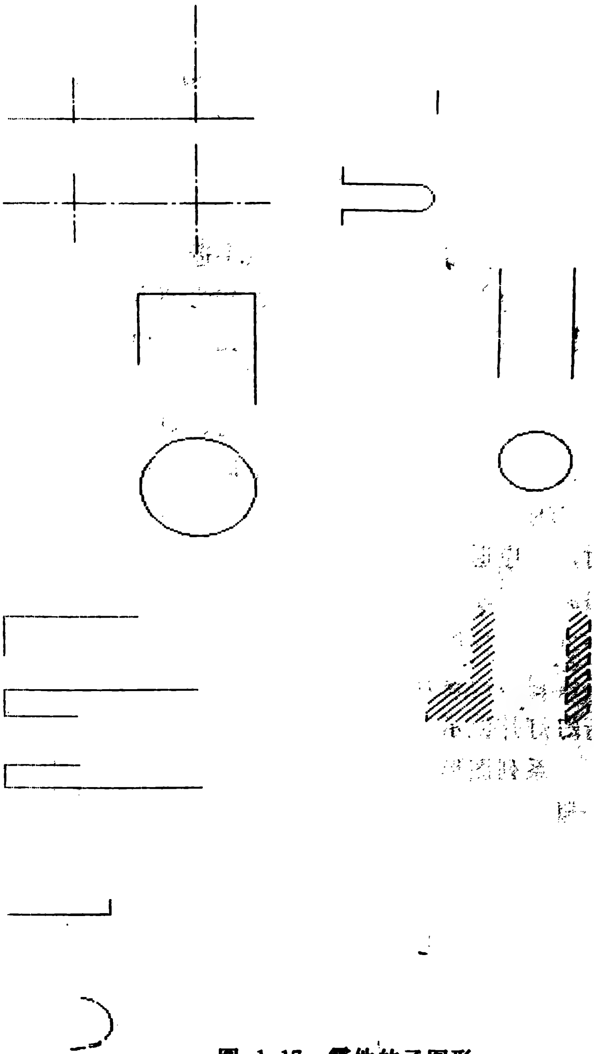


图 1.17 零件的子图形

问题。

图1.16是一个机械零件的主俯视图，要求顺序演示这个主俯视图的作图步骤。

· 按照专业知识，分七步来完成图1.16，由此分解每一个子图形，共有 7 个子图形，见图1.16。按照各子图形的要求，对于不同的部分分别建立相应的子造型，比如图1.17(a)，需建立 3 个子造型，图1.17 (b,c,d,e,f)均需建立 2 个子造型，而图1.17 (g)作为一个造型来处理。

· 由图1.17，根据图形数据交互生成程序GP1407生成 7 幅图形，最好按执行顺序完成上述图形的生成。

①显示程序（见GP 1412）

```
5   REM    GP1412
6   PRINT  CHR$ (4);"BLOAD 1412-1"

20   POKE 232,0: POKE 233,96
30   FOR I = 1 TO 8
40   READ X(I),Y(I)
50   NEXT I
60   HGR : HCOLOR= 3
70   SCALE= 1: ROT= 0
80   HOME : VTAB 23
90   PRINT "A---- All Graphics     E
      ----End"
100   PRINT "S    Drawing Step
      X----Xdraw"
110   GET A$
120   IF A$ =  CHR$ (65) THEN  GOSUB
      170
130   IF A$ =  CHR$ (88) THEN  HGR

140   IF A$ =  CHR$ (69) THEN  GOSUB
      380: END
```

```
150   IF A$ =  CHR$ (83) THEN  GOSUB
      210: GOTO 80
160   GOTO 110
165   END
170   FOR I = 1 TO 8
180   DRAW I AT X(I),Y(I)
190   NEXT I
200   RETURN
210   HPLOT 136,0 TO 136,159
220   HPLOT 134,0 TO 134,159
230   HOME : VTAB 23
240   PRINT "Please Press RETURN C
      ontinue......"
250   GET A$
260   IF A$ <  >  CHR$ (13) THEN 2
      50
270   DRAW 1 AT X(1) + 142,Y(1): DRAW
      3 AT X(3) + 142,Y(3)
280   GET A$
290   IF A$ <  >  CHR$ (13) THEN 2
      80
300   DRAW 2 AT X(2) + 142,Y(2)
305   FOR I = 4 TO 8
310   GET A$
320   IF A$ <  >  CHR$ (13) THEN 3
      10
330   DRAW I AT X(I) + 142,Y(I)
340   NEXT I
350   RETURN
380   TEXT : HOME
390   VTAB 12: PRINT "
         Good  Bye"
400   RETURN
410   DATA  100,2,102,124,38,80,6
      0,49,38,140,39,133,100,124,7
      6,5
```

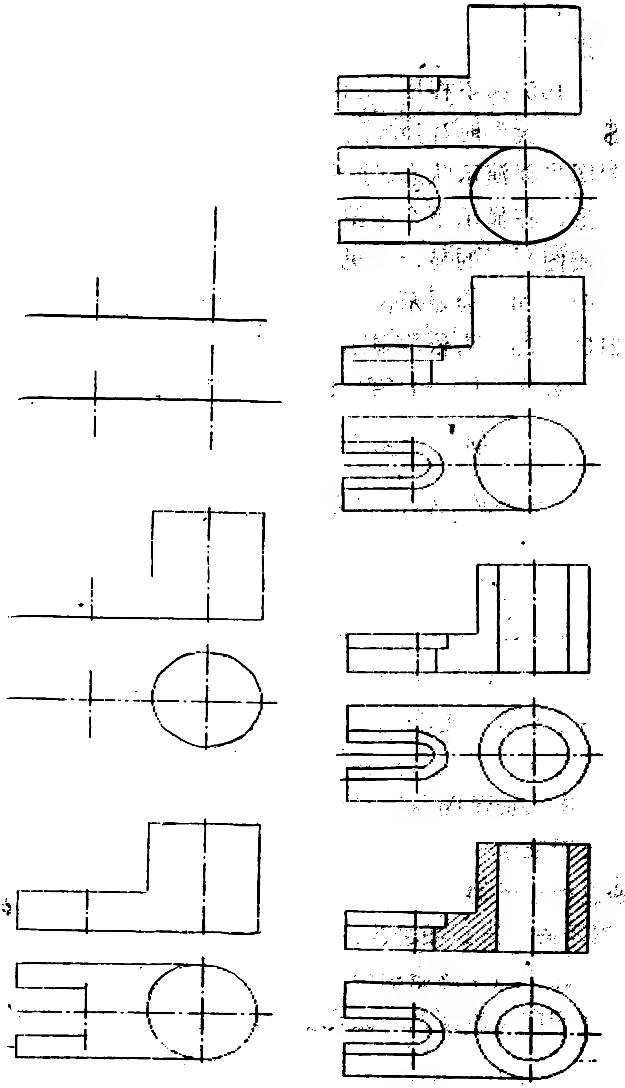


图 1.18 零件制图步骤的演示

②程序说明

5 —165 为主程序，当A $ = “A” 时，显示图1.16；当A $ = “X” 时，清除屏幕显示的图形；当A$ =“S”时，进入制图步骤演示状态，这时依次按回车键，顺序显示图1.17各子图形，在显示下个子图时而不擦掉原来显示的图，共分七步完成图1.16的显示，见图1.18。

170 —200 为显示图1.16的子程序。

210 —350 为依次显示图1.17子图形的子程序。

远行程序G P 1411需要图1.17的造型表。

第五章　动态图形的组织

第四章从数据结构的角度谈到了静止图面的组织，并给出了相应的工具软件。本章主要介绍用高级语言来组织动态图形基本方法及有关动态图形程序的设计技巧。最后，探讨一下从数据结构的角度如何生成动态图形。

第一节　动态图形的显示原理

本节主要介绍动态图形的显示原理，具体实现，高解析度图形的两页及消除闪烁的换页显示。

一、单页屏幕上显示动态图形的方法

所谓单页屏幕，就是指绘图时只在一个高解析度屏幕上进行，或者只使用第一页（HGR），或者只使用第二页（HGR2）。那么怎样在一个固定的屏幕上产生动态图形呢？首先来分析大家都熟悉的例子——电影。

电影，是通过把胶片上一个个不动的画面，连续地投射到银幕上面形成的，观众之所以能够看到一个连续运动的画面，是因为在极短的时间里令第一个画面消失，而立即显示出稍微运动了一点的新的画面。由电影的形成过程可以设想，如果能在微机的屏幕上做到令一个图形消失，然后在新的位置上以很快的速度画出这幅图形或者是变化了的图形，则也同样会使人感觉到图形在运动。

由上述分析可知，产生动画的关键就是要不断地擦去旧

图形，画出新图形。关于图形显示前面几章已经介绍了许多方法，下面主要讨论擦图问题。交互式图形数据生成程序GP1407，对图形的动态牵引，实际上是一个由人控制的动画，每移动一个距离，老图形就被擦去，并在新的位置上显示原来的图形。

1．使用XDRAW指令擦图

由于 XDRAW 指令的功能，是以图面上的补色来画图的，因此，第一次使用这个指令画图时，不管其背景是什么颜色，都能画出一个与背景互成补色的图形。但是如果再使用该指令，在原来基础上重画图形，则因为它又将按着原来的补色画图，故所画之处又还原为原来背景颜色，于是就把原来的图形擦掉了。例如程序GP1501可以将一个“中”字从屏幕一端移动到另一端。该程序是一个自带造型表，其造型表中的造型为“中”字。

```
5   REM   GP1501
7   HGR2
10   FOR I = 1 TO 20
15   READ A: POKE 24575 + I,A
20   NEXT
30   POKE 232,0: POKE 233,96
40   SCALE= 1: ROT= 1
50   FOR I = 20 TO 259 STEP 2
60   XDRAW 1 AT I,70
70   FOR J = 1 TO 100: NEXT J
110   XDRAW 1 AT I,70
120   NEXT
125   END
130   DATA  1,0,4,0,36,45,45, 45,
      5 4,63,63,127,9,36,36,54,54,
      54,54,0
```

70句为延时语句，动画是由50—120 句实现的。

2．设定补色来擦去图形

由于 XDRAW 指令只能擦去以造型法画出的图形，对于以点线法画的图就无能为力了。因而，如果图形中运动部分是用点线法或两种方法交互使用时，就只能用“设定补色”的方法来实现擦图。对于单颜色的荧光屏，只有黑、白两种互为补色的颜色，使用此法尤为方便。在图形数据交互生成程序GP1407中，动态牵引造型就是用设定补色来实现的。以后章节均以设定补色来研究动态图形，程序GP1502给出了具体的使用方法。

```
5   REM    GP1502
10   HGR2
20   FOR X = 0 TO 190 STEP 4
25  Y = X
30   HCOLOR= 3: GOSUB 110
50   FOR J = 1 TO 100: NEXT
60   HCOLOR= 0: GOSUB 110
70   NEXT
80   END
110   HPLOT X,Y TO 190 - X,Y
120   HPLOT  TO 190 - X / 2,190 -
      Y / 2
140   HPLOT  TO X,Y
150   RETURN
```

程序GP1502中，110—150 子程序是画三角形的程序，其值由X，Y值来确定。20—70句实现这个三角形由大变小，然后由小变大的动态图形，是不断交替的画与抹，30句按白色画一个三角形，60句按黑色抹去这个三角形，至于画图和擦图时使用的颜色并不是一成不变的，其颜色互补关系见表1.3，如对于白屏幕画黑色图形的情况，其使用的颜色对上面

的程序需要颠倒一下顺序。

二、双页屏幕显示动态图形

上面介绍了在单页状态下显示动态图形方法，并给出了两个简单的实例程序，细心的操作者会发现这两个图形是在闪动状态下运动的，通过在擦与画的过程中间增加延时时间，可以减少闪动状态，但是最终还是不能消除闪动。这是因为高级语言执行速度慢，新旧画面的更替速度不够快而造成的。实际上，擦图与画图过程完全相同，因而其占用的时间都将随着图形的复杂化而增长，换句话说，不可能在人的视觉停留范围内实现图形的更替，其结果便造成图形闪动的感觉。为了减轻图形的闪动，可在画与擦之间人为增加停留时间；为了消除图形的闪动，可设置两个屏幕图形缓冲区，同时在两个屏幕上画图，并通过两页的瞬间互换，而显示一个高质量的动态图形，这种方法又称为换页图形显示法。

我们知道中华学习机的 BASIC 语言有两个高解析度屏幕，可以用HGR和HGR2来设定它们，先在第一页上画出一个处于某一位置的图形，而后接着在看不见的第二页上画下一个位置的图形，然后在一瞬间隐去第一页，显示出第二页，就能够得到一个不具有闪动的动态图形了。前一页隐去后，还要实现擦旧图画新图的过程，以备下次换页时使用，只是这一擦与画的动作是在幕后进行的，我们看不见，也不占用新旧画面更替的时间，如此变更图形显示的页次，就能够显示出一个连续运动的图形，但如何实现换页，这还得从微机显示屏幕的几个软开关及有关地址指令说起。

1. 控制绘图页次的软开关

中华学习机内存中划有 4 个特定的显示区：文本、低解

析度图形显示的第一页、第二页、高解析度图形显示的第一页和第二页缓冲区。除了用TEXT、GR、HGR和HGR2的 BASIC 指令可以方便地选用它们之外，还可以通过四组软开关选用它们来控制显示状态。

这四组软开关是：

① POKE —16303，0 文本型显示
   POKE —16304，0 图形显示

② POKE —16300，0 显示第一页
   POKE —16299，0 显示第二页

③ POKE —16298，0 低解析度一、二页显示
   POKE —16297，0 高解析度一、二页图形显示

④ POKE —16302，0 高、低解析度图形全屏幕
   POKE —16301，0 图形⊕、文本混合显示

由上述的四组软开关，可以根据需要，单独地或是组合起来使用，例如选择软开关的组合为：

POKE —16304，0 图形显示
POKE —16297，0 高解析度图形
POKE —16300，0 第一页显示
POKE —16302，0 全屏幕显示

就能使得高解析度图形的第一页全屏被显示出来而不致于将它前面所存的图形清除掉。而下面的软开关组合为：

POKE —16304，0 图形显示
POKE —16297，0 高解析度图形
POKE —16299，0 第二页显示

就能把高解析度第二页设成全屏幕，而不清除它前面所存的图形。

2．换页显示的软开关控制

有了上述软开关，我们就能从一个高解析度图形页的图形显示，一下子跳到另一个图形页的显示。

从第一页进入第二页用POKE －16299，0

从第二页进入第一页用POKE －16300，0

因为换页显示的时间极短，一般眼睛觉察不到这一更替过程，可以生成卡通式的高质量的动态图形。

3．在看不见的高解析度图形上画图

前述谈到的图形的交互生成一系列工具，图形均是画在所设定的图形页上。当图形画在第一页，实际上是在＄2000—＄3FFF图形缓冲区送入相应的数据。当图形画在第二页上，实际上是向＄4000—＄5FFF的图形缓冲区内填入相应的数据，图形第一页与第二页屏幕与相应的缓冲区一一对应，至于是第一页还是第二页，还要看专用的内存地址＄E6（十进制为230）中的数值而定，如果＄E6中存放数值为32时，所有的图形都将画在第一页内；而如果＄E6中存放的数值为64，则所有的图形均画在第二页内。

掌握了地址＄E6中数值对高解析度图形页的制约关系，就可以人为地改变＄E6中的数值，从而达到向某一个高解析度图形页画图的目的。

设定第一页和第二页时，用下述命令：

HGR2：HGR：POKE －16302，0

当在第二页画面上画图并显示第一页，可用命令：

POKE －16300，0：POKE230，64

当在第一页画面上画图并显示第二页图形，可用命令：

POKE －16299，0：POKE230，32

以上控制画图页次的命令，可以在立即状态下使用，也可以灵活地编在自己的程序中。程序G P 1503可以将"中"字不闪烁地从屏幕的左端移到右端、采用的是换页图形显示。

① 源程序

```
5   REM    GP1503
10   HGR2 : HGR : POKE  - 16302,0
20   FOR I = 1 TO 20
30   READ A: POKE 24575 + I,A
40   NEXT
50   POKE 232,0: POKE 233,96
60   SCALE= 4: ROT= 1
70   FOR I = 20 TO 250 STEP 2
80   POKE    16300,0: POKE 230,64
90   HCOLOR= 0: DRAW 1 AT I,70
100   HCOLOR= 3: DRAW 1 AT I + 2,70
110   POKE  - 16299,0: POKE 230,32
120   HCOLOR= 0: DRAW 1 AT I,70
130   HCOLOR= 3: DRAW 1 AT I + 2,70
140   NEXT
150   END
160   DATA   1,0,4,0,36,45,45, 45,5
      4,63,63,127,9,36,36,54,54,54,5
      4,0
```

② 程序说明

10句为设置高解析度两页，均为全屏幕。

20—40为将"中"字造型表送进 24576 为起始处的内存。

70— 140 为换页动态图形显示，将 "中" 从左端移动到右端。

80句为显示第一页并在第二页上画图。

110句为显示第二页并在第一页上画图。

第二节　动态图形的设计技巧

计算机模拟包括两个方面：定量模拟和定性模拟，前者事先不知变化后的真实现象，通过一定的计算分析，把其真实现象用动态图形给显示出来，比如超速飞机飞行时，飞机两翼的挠曲等，一个零件加工现象的模拟等等。后者知其结果，通过模拟，可以形象化反应其结果，它的典型应用就是在机算机辅助教学方面，通过模拟一项课程实验，如帕斯卡原理、物体的自由落体运动、电荷在磁场运动的轨迹等等，或者是教科书上不易理解的概念、现象，通过夸张的手段反应其运动过程，使不易理解的概念形象化、直观化。

本节主要用实例介绍定性模拟的动态图形制作方法及技巧，主要包括两方面的内容，其一、能够用一定函数关系描述的动态图形的制作方法。其二、不能用一定函数关系来描述，而可以用夸张的手段反应其内在运动过程的动态图形的制作方法。

一、单个造型的任意方向移动

这里被移动的造型为一个小球，如果要移动其它形状的造型，则只要用前述的工具软件建立相应的造型来替换小球的造型，也可实现相似的运动。

1．作水平运动的造型

图形作水平移动时，只要保持Y坐标值不变，改变X坐标。程序G P 1504可以将小球从左边等速地移动到右边。

① 源程序

```
5  REM    GP1504
7  DATA    1,0,4,0,9,228,63,30,54,
```

```
      14,45,12,4,0
8  FOR I = 1 TO 14
9  READ A: POKE 24575 + I,A: NEXT

10 V = 40
15  HGR2 : ROT= 0: SCALE= 1
20 K1 = 232:K2 = 233: HCOLOR= 3
30  POKE K1,0: POKE K2,96
40  FOR T = 0 TO 6 STEP 0.3
50 X = V * T
60  HCOLOR= 3: GOSUB 200
80  HCOLOR= 0: GOSUB 150
90  NEXT
95  HCOLOR= 3: GOSUB 150
100  END
150  DRAW 1 AT X,20
160  RETURN
200  HPLOT X,0
230  DRAW 1 AT X,20
250  RETURN
```

② 程序说明

5 — 100 为主程序，本程序是一个自带造型表的程序。由 7 — 9 句将造型数据存到＄6000为起始处。40—90句为实现造型的移动，一画一抹，即新旧画面更替，还可以留下小球运动的轨迹。50句为小球作等速移动的函数式子。

输入并运行程序G P 1504可以看出，造型的移动有闪动，并很快地从左端移动到右端。人为地在新旧画面更替时加延时语句，比如加一条：

70 FOR J = 1 TO 100:NEXT J

则可以减少闪动。程序G P 1505采用换页显示，完全可以避免图形的闪动。

```
5  REM   GP1505
7  DATA   1,0,4,0,9,228,63,30,54,
     14,45,12,4,0
8  FOR I = 1 TO 14
9  READ A: POKE 24575 + I,A: NEXT

10 V = 40
15  HGR2 : HGR : POKE  - 16302,0
17  SCALE= 1
20 K1 = 232:K2 = 233: HCOLOR= 3
30  POKE K1,0: POKE K2,96
35 S = 0.I * V
40  FOR T = 0 TO 6 STEP 0.1
50 X = V * T
55  POKE  - 16300,0: POKE 230,64
60  HCOLOR= 0: GOSUB 200
65  HCOLOR= 3: GOSUB 150
70  POKE  - 16299,0: POKE 230,32
75  HCOLOR= 0: GOSUB 200
80  HCOLOR= 3: GOSUB 150
90  NEXT
95  HCOLOR= 3: GOSUB 150
100  END
150  DRAW 1 AT X + S,20
160  HPLOT X,0
170  RETURN
200  DRAW 1 AT X,20
230  RETURN
```

2. 造型的垂直运动

在新旧画面更替时，只要固定X坐标，改变Y坐标值，就能实现造型的垂直运动。程序G P 1506可以模拟小球作垂直下抛运动。

① 源程序

```
5  REM   GP1506
```

```
7  DATA   1,0,4,0,9,228,63,30,54,
      14,45,12,4,0
8  FOR I = 1 TO 14
9  READ A: POKE 24575 + I,A: NEXT

15  HGR2 : SCALE= 1: ROT= 0
20 K1 = 232:K2 = 233: HCOLOR= 3
30  POKE K1,0: POKE K2,96
40  FOR T = 0 TO 6 STEP 0.3
50 Y = 4.9 * T * T
60  HCOLOR= 3: GOSUB 200
70  FOR I = 1 TO 400: NEXT
80  HCOLOR= 0: GOSUB 150
90  NEXT
95  HCOLOR= 3: GOSUB 150
100  END
150  DRAW 1 AT 20,Y
160  RETURN
200  HPLOT 0,Y
230  DRAW 1 AT 20,Y
240  RETURN
```

② 程序说明

5—100为主程序，类似于上面的程序，不同的是50句，该句为造型作自由落体的函数式子。

如果用下面的语句替换程序中相应的语句，则可慢速演示小球的垂直上抛运动。

```
40  FOR  T = 6  TO  O  STEP —0.3
```

3. 造型作抛物线移动

如果小球不作垂直运动，而是将小球水平投掷，在重力作用下，小球的运动轨迹为抛物线，这样的程序如何设计。

① 构造数学模型

按照物理学的知识，水平投掷运动的小球，在重力的作

用下，其轨迹按下列公式确定：

$$X = X_0 + vt$$

$$Y = y_0 + \frac{1}{2}gt^2$$

② 源程序（见G P 1507）

```
5   REM   GP1507
6   DATA      1,0,8,0,0,0,0,0
7   DATA    9,228,63,30,54,14,45,1
      2,4,0
8   FOR I = 1 TO 18
9   READ A: POKE 24575 + I,A: NEXT

19   HGR2 :V = 50
20 K1 = 232:K2 = 233
25   HCOLOR= 3: SCALE= 1
30   POKE K1,0: POKE K2,96
40   FOR T = 0 TO 5 STEP 0.2
50 X = V * T:Y = 4.9 * T * T
60   HCOLOR= 3: GOSUB 200
70   FOR J = 1 TO 200: NEXT
80   HCOLOR= 0: GOSUB 150
90   NEXT
95   HCOLOR= 3: GOSUB 150
100   END
150   DRAW 1 AT X + 10,Y + 10
170   RETURN
200   HPLOT X + 10,Y + 10
230   DRAW 1 AT X + 10,Y + 10
260   RETURN
```

③ 程序说明

5—100句为主程序，5—9句将小球造型表存入＄6000为

起始处，40—90句为新旧画面的更替，小球运动的轨迹由50句来控制，70句为新旧画面更替时的延时循环。

如果在屏幕中增加辅助信息，如图1.19则可以模拟带正电的质点受磁场力作用而发生偏移的定性模拟，相应的程序为GP 1508。

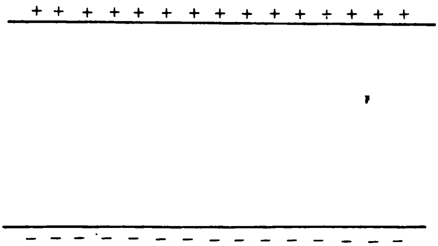


图 1.19

① 源程序

```
5   REM   GP1508
6   DATA    3,0,8,0,18,0,26,0
7   DATA  9,228,63,30,54,14,45,12,
      4,0
8   DATA  45,45,221,35,52,54,54,0,
      255,45,45,5,0
9   FOR I = 1 TO 31
10   READ A: POKE 24575 + I,A: NEXT

15   HGR2 : SCALE= 1
20 K1 = 232:K2 = 233: HCOLOR= 3
30   POKE K1,0: POKE K2,96
32   HPLOT 3,10 TO 260,10
33   HPLOT 0,160 TO 260,160
34   FOR I = 1 TO 260 STEP 10
```

```
36  DRAW 2 AT 3 + I,6: NEXT
37  FOR I = 1 TO 260 STEP 10
38  DRAW 3 AT 4 + I,165: NEXT
40  FOR T = 0 TO 4.2 STEP 0.2
50 X - 65 * T:Y = 4.9 * T * T
60  HCOLOR= 3: GOSUB 200
65  FOR I = 1 TO 400: NEXT
80  HCOLOR= 0: GOSUB 150
90  NEXT
95  HCOLOR= 3: GOSUB 150
100  END
150  DRAW 1 AT X,Y + 80
160  RETURN
200  HPLOT X,Y + 80
230  DRAW 1 AT X,Y + 80
240  RETURN
```

② 程序说明

5—100为主程序，9—10将三个造型的造型表存入内存＄6000为起始处，第一个造型为小球，第二个造型为“＋”，第三个造型为“－”，20—38句为画图1.19；其它类似于上面的程序说明。

4．模拟球的弹跳运动

如何模拟一个小球从一定高度掉到地面后的运动。

① 构造数学模型

由物理学可知，小球从某一高度掉到光滑地面上的运动为一阻尼运动，其轨迹可以用下面式子：

Y＝H＊SIN(X＋D)＊EXP(－K＊X)

来表示。

② 源程序（见GP1509）

```
5   REM   GP1509
6   DATA   1,0,4,0,9,228,63,30,54,
      14,45,12,4,0
8   FOR I = 1 TO 14
9   READ A: POKE 24575 + I,A
10   NEXT
15 PI = 3.14159:W = PI / 40:D = 9
      0 * PI / 180
17 H = 140:K = 0.01
20   HCOLOR= 3: SCALE= 1
30   POKE 232,0: POKE 233,96
40   HGR2
45   HPLOT 10,160 TO 276,160
50   FOR X = 0 TO 270 STEP 2
60 Y = H *  SIN (W * X + D) *  EXP
      ( - K * X)
70 Y = 160 -  ABS (Y) - 3
80   HCOLOR= 3: GOSUB 200
90   FOR J = 1 TO 100: NEXT
100   HCOLOR= 0: GOSUB 250
110   NEXT
120   HCOLOR= 3: GOSUB 250
140   END
200   DRAW 1 AT X,Y
205   HPLOT X,Y
210   RETURN
250   DRAW 1 AT X,Y
260   RETURN
```

③ 程序说明

5—140为主程序，取衰减系数K=0.01，6—10句将小球的造型数据存入＄6000为起始处，45句画一水平线表示地面，50—110句为模拟小球的弹跳运动，60句求小球弹跳的高度，

70句实现坐标转换，以地面（Y＝160处）向上为Y轴方向。

二、多个造型的同时移动

上面介绍的一些例子，只是一个造型的移动，在计算机定性模拟中，或者是动画片的制作，往往是多个局部图形按各自规律在屏幕上运动。这类程序如何设计？

1．多个造型同时运动的原理

大家知道，一个造型的动画就是在一个位置上显示它，然后在新的位置上显示它，同时擦去老位置上的造型，给人感觉到这个造型在运动。要想让多个造型同时按各自规律运动，与单个造型运动具有相同的原理，只是在新老图面更替时，均是多个造型，即在一个位置上画出多个造型，然后擦除它并在新的位置上显示同样多个造型，就会获得多个造型在同时运动。

2．按照各自规律运动的多个造型

向下斜抛一个物体，要求在一个屏幕上同时显示三个造型的运动，物体的运动及在X，Y轴上的投影运动，并分别留下相应的轨迹，可采用单页动态图形显示方法实现。

① 构造数学模型

由图1.20可知，向下斜抛一个物体的轨迹为一抛物线，在X轴上投影运动为等速运动，在Y轴上为垂直下抛的加速运动，其合成运动由下式参数方程给出：

$$X = X_0 + (v_0 \cos\theta) t$$

$$Y = Y_0 + (v_0 \sin\theta) t + \frac{1}{2} g t^2$$

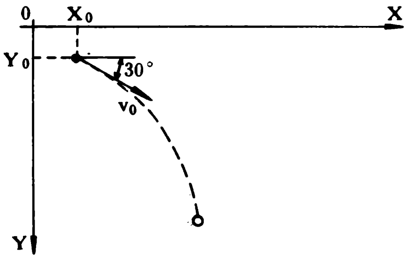


图 1.20 斜抛运动

② 源程序（见GP1510）

```
5   REM    GP1510
6   DATA        3,0,8,0,18,0,22,0
7   DATA      9,228,63,30,54,14,45,1
      2,4,0
8   DATA      45,63,47,0,36,54,54,38
      ,0
9   FOR I = 1 TO 27
10   READ A: POKE 24575 + I,A: NEXT

12 TH = 30 * 3,14159 / 180:V0 = 5
      0
15   HGR2
20 K1 = 232:K2 = 233
25   HCOLOR= 3: SCALE= 1
30   POKE K1,0: POKE K2,96
40   FOR T = 0 TO 4 STEP 0.1
50 X = 10 + V0 *  COS (TH) * T:Y =
      10 + V0 *  SIN (TH) * T + 4.
      9 * T * T
60   HCOLOR= 3: GOSUB 200
70   FOR J = 1 TO 200: NEXT
```

```
80   HCOLOR= 0: GOSUB 150
90   NEXT
95   HCOLOR= 3: GOSUB 150
100  END
150  DRAW 1 AT X,Y
155  DRAW 2 AT X,10
160  DRAW 3 AT 10,Y
170  RETURN
200  HPLOT X,Y
210  HPLOT X,8
220  HPLOT 8,Y
230  DRAW 1 AT X,Y
240  DRAW 2 AT X,10
250  DRAW 3 AT 10,Y
260  RETURN
```

③ 程序说明

5—100句为主程序，9—10句将3个造型的造型表存入$6000为起始处，40—90句为动态图形的描述语句，50句为函数式子,用来确定被移动对象的轨迹,10句与80句为更替画面，70句为延时循环。

150—170子程序为擦图模块，一号造型为小球，作抛物线运动，二号造型为小球在X轴上投影运动，三号造型为小球在Y轴的投影运动。

200—260子程序为画图模块,200句留下一号造型的运动轨迹，210句留下二号造型的运动轨迹，220句留下三号造型的运动轨迹。

3个造型的起点均为各自造型的几何中心。

3．模拟水压机原理（帕斯卡原理）

模拟水压机动作原理，要求能够显示压力及活塞受压状

态及分布，见图1.21。

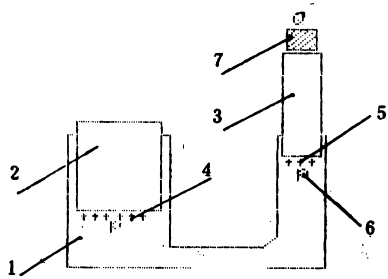


图 1.21 水压机原理示意图

1. 油缸；2. 大活塞；3. 小活塞；4、5. 压力；6. 重物。

① 构造数学模型

大家知道水压机的两个活塞的行程与其面积成反比关系，即：

$$S_1 \cdot l_1 = S_2 \cdot l_2$$

$$\frac{\pi d_1}{4} \cdot l_1 = \frac{\pi d_2}{4} \cdot l_2$$

$$\frac{l_1}{l_2} = \left(\frac{d_2}{d_1}\right)^2$$

若取 $d_2/d_1 = 2$，则小活塞下降 4 个坐标单位，大活塞上升一个坐标单位。

② 图表建立

由图1.21为了编程方便，按顺序全部用造型表来完成，第 1 个造型表示油缸，第 2 个造型表示大活塞，第 3 个造型表示小活塞，第 4 和第 5 个造型表示压力，第 6 个造型表示字符 p，第 7 个造型表示重物，造型表见 GP1511-1，除了

油缸之外，在模拟水压机原理过程中，同时运动有6个造型。在完成各造型之后，由图型数据交互生成工具建立图1.21，并确定各造型初始位置坐标。

③ 源程序（见GP1511）

```
5  REM   GP1511
7  PRINT  CHR$ (4);"BLOAD GP1511-
     1,A$6000"
10  DIM Y1(20),Y2(20)
20  POKE 232,0: POKE 233,96
40  HGR : POKE  - 16302,0
50  HGR2 : SCALE= 1
60  HCOLOR= 3: GOSUB 700
70  POKE 230,32: GOSUB 650
90  POKE 230,64: GOSUB 650
200  FOR I = 1 TO 12
210  POKE  - 16300,0: POKE 230,64

230  HCOLOR= 0:J = I
240  GOSUB 800: GOSUB 870
260  HCOLOR= 3:J = I + 1
270  GOSUB 800: GOSUB 870
290  POKE  - 16299,0: POKE 230,32

310  HCOLOR= 0:J = I
320  GOSUB 800: GOSUB 870
340  HCOLOR= 3:J = I + 1
350  GOSUB 800: GOSUB 870
370  NEXT I
375  HCOLOR= 0: POKE 230,32: GOSUB
     870
378  POKE 230,64: GOSUB 870
380  FOR I = 1 TO 50: NEXT I
470  FOR I = 14 TO 2 STEP  - 1
```

```
480  POKE  - 16300,0
490  POKE 230,64
500  HCOLOR= 0:J = I
510  GOSUB 800: GOSUB 860
520  HCOLOR= 3:J = I - 1
530  GOSUB 800: GOSUB 860
540  POKE  - 16299,0: POKE 230,32
560  HCOLOR= 0:J = I
570  GOSUB 800: GOSUB 860
580  HCOLOR= 3:J = I - 1
590  GOSUB 800: GOSUB 860
600  NEXT I
610  HCOLOR= 0: POKE 230,32: GOSUB
     860
620  POKE 230,64: GOSUB 860
630  GOTO 200
640  END
650  DRAW 1 AT 80,80
660  RETURN
700  FOR I = 1 TO 20
710 Y2(I) = 30 + 4 * I
720 Y1(I) = 80 - I
730  NEXT
740  RETURN
800  DRAW 2 AT 81,Y1(J)
810  DRAW 3 AT 172,Y2(J)
820  DRAW 4 AT 84,Y1(J) + 45: DRAW
     5 AT 173,Y2(J) + 56
830  DRAW 6 AT 96,Y1(J) + 54
840  DRAW 6 AT 178,Y2(J) + 66
850  RETURN
860  DRAW 7 AT 92,Y1(J) - 13
865  RETURN
870  DRAW 7 AT 173,Y2(J) - 11
880  RETURN
```

④ 程序说明

5—640句为主程序，设置两个数组Y1(I)与Y2(I)分别存放大活塞及小活塞移动路径，该路径由60句实现，即小活塞下降4个坐标单位，大活塞上升一个坐标单位。70与90句在两个图形页上画出油缸，200—370句为模拟小活塞上承受的重物向下移动，大活塞向上移动，380—600句为模拟大活塞上承受重物向下移动，小活塞向上移动，要停止程序运动，请按复位键。

700—740子程序为计算大小活塞的移动坐标，并分别存入Y1(I)及Y2(I)数组中。

800—850子程序为画大小活塞及压力模块。

860—865子程序为在大活塞上画出重物模块。

870—880子程序为在小活塞上画出重物模块。

```
GP1511-1

*6000.6250

6000- 0A 07 20 00 00 01 57 01
6008- A2 01 C7 01 DB 01 EA 01
6010- FF FF FF FF FF FF FF FF
6018- FF FF FF FF FF FF FF 00
6020- 3F 37 36 36 36 36 36 36
6028- 36 36 36 36 36 36 36 36
6030- 36 36 36 36 36 36 36 36
6038- 36 36 36 36 36 36 36 36
6040- 36 36 36 36 36 2D 2D 2D
6048- 2D 2D 2D 2D 2D 2D 2D 2D
6050- 2D 2D 2D 2D 2D 2D 2D 2D
6058- 2D 2D 2D 2D 2D 2D 2D 2D
```

```
6060- 2D 2D 2D 2D 2D 2D 2D 2D
6068- 2D 2D 2D 2D 2D 2D 2D 2D
6070- 2D 2D 2D 2D 2D 2D 2D 2D
6078- 2D 2D 2D 2D 2D 25 24 24
6080- 24 24 24 24 24 24 24 24
6088- 24 24 24 24 24 24 24 24
6090- 24 24 24 24 24 24 24 24
6098- 24 24 24 24 24 24 24 24
60A0- 24 3F DF DB DB DB DB 1B
60A8- 3F 36 36 36 36 36 36 36
60B0- 36 36 36 36 36 36 36 36
60B8- 36 36 36 36 36 36 36 36
60C0- 36 36 36 36 1E 1E 1E 1E
60C8- 1E 1E 3F 3F 3F 3F 3F 3F
60D0- 3F 3F 3F 3F 3F 3F 3F 3F
60D8- 3F 3F 3F 3F 3F 3F 27 24
60E0- 24 24 24 24 24 24 24 24
60E8- 24 24 24 24 24 24 24 24
60F0- 24 24 24 24 24 24 24 24
60F8- 24 24 24 24 3C 3F 07 00
6100- 26 34 36 36 36 36 36 36
6108- 36 36 36 36 36 36 36 36
6110- 36 36 36 36 36 36 36 2D
6118- 2D 2D 2D 2D 2D 2D 2D 2D
6120- 2D 2D 2D 2D 2D 2D 2D 2D
6128- 2D 25 24 24 24 24 24 24
6130- 24 24 24 24 24 24 24 24
6138- 24 24 24 24 24 24 24 24
6140- 24 3F 3F 3F 3F 3F 3F 3F
6148- 3F 3F 3F 3F 3F 3F 3F 3F
6150- 3F 3F 3F 37 36 36 00 33
6158- 36 36 36 36 36 36 36 36
6160- 36 36 36 36 36 36 36 36
6168- 36 36 36 36 36 36 36 36
```

```
6170-  36  36  2D  2D  2D  2D  2D  2D
6178-  2D  2D  25  24  24  24  24  24
6180-  24  24  24  24  24  24  24  24
6188-  24  24  24  24  24  24  24  24
6190-  24  24  24  24  24  24  3F  3F
6198-  3F  3F  3F  3F  3F  3F  37  36
61A0-  36  00  26  F4  2D  95  1B  4C
61A8-  49  26  24  1E  2D  4D  25  36
61B0-  26  2C  4D  09  F4  2D  37  66
61B8-  09  09  24  3E  2D  37  66  49
61C0-  21  34  2F  3D  36  06  00  21
61C8-  3E  2D  37  66  49  21  34  2F
61D0-  3D  36  64  49  21  3E  2D  37
61D8-  76  01  00  24  24  24  2E  0C
61E0-  75  36  1E  27  BB  4A  11  49
61E8-  01  00  36  36  36  2E  2D  2D
61F0-  2D  2D  2D  2D  24  24  24  24
61F8-  24  3C  3F  3F  3F  3F  3F  3F
6200-  36  36  0C  0C  0C  2C  2D  F5
6208-  1E  1E  1E  1E  1E  1E  76  0C
6210-  0C  0C  0C  0C  0C  0C  0C  2C
6218-  2D  1E  F2  1E  1E  1E  1E  1E
6220-  1E  1E  2E  65  0C  0C  0C  0C
6228-  0C  0C  94  52  1E  1E  1E  4E
6230-  C9  24  24  24  24  24  C4  18
6238-  D8  DB  DB  20  24  64  2D  0E
6240-  36  F6  1C  94  3A  4F  0D  4D
6248-  49  01  00  00  00  00  00  00
6250-  49
```

三、按预定路径移动造型

上面介绍了单个造型以及多个造型的运动，其运动路径均能用函数式子来描述，在计算机辅助教学中，为了形象、直观，把知识用其过程来描述，如减速器拆装实验模拟，电子、电流走向的模拟，液压传动阀的动作模拟。对这样一类

过程的描述，常需要造型按预定的路径来移动，往往这样的路径用函数式子描述较困难或不能用函数式子来描述，但可以按预定要求，通过方格纸或工具软件找出路径的一系列坐标。下面就来谈谈这方面动态图形的制作方法。

1. 一般步骤

① 运动过程的描述

根据被模拟对象，运用有关专业知识来刻划其功作过程及相应的要求，有目的、要求、具体内容和实施办法等。

② 图形的制作

根据第一步确定制作的图面，对静正的画面，可以用节点法来完成，对运动对象可以用造型来实现。并由工具软件生成相应的图面及造型表。

③ 动画位置的确定

由动作过程，在方格纸上画出其相应的轨迹，每个需要运动的造型都有一个运动的轨迹，轨迹坐标直接在方格纸上截取。

根据动作过程，由图形数据交互生成工具GP 1407。每一个运动的造型，沿着轨迹复制一幅图形。

④ 程序设计

根据动作过程及相应要求，针对已经建立的图形、图表和动画路径坐标，写出执行细节或画出框图，完成编程及调试工作。

下面介绍一下实例，看看如何设计这样一类的程序。

2. 带图形汉字的通用演示程序

下面以通用造型软件包的引导程序为例，见图1.22，介绍带图形汉字的通用演示程序设计。

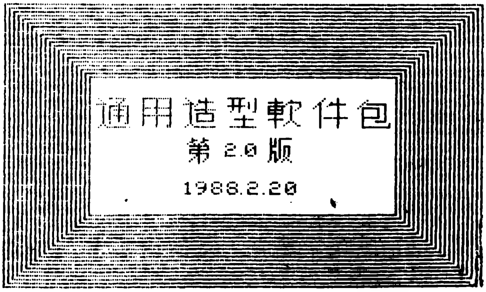


图 1.22　带汉字的演示屏幕

① 过程描述

图1.22为通用造型软件包的引导程序最终显示的画面，按照什么样的过程来动态表演，最终显示图1.22，不同的人会设计不同的过程描述，这里只是其中的一种。首先随机旋转动态演示通、用、造、型、软、件和包，依次在屏幕中间位置累加一行"通用造型软件包"，其次由下往上移动"1988．2.20．第 2.0版"到其屏幕的适当位置，最后矩形扩展并留下相应的矩形。图1.22为最终显示的图形，在运动过程中，要求按任意键立即结束或进入主菜单程序。

② 图形制作

见图1.22因为每个汉字及字符均需要动态演示，因此，全部用造型，造型的顺序为：通、用、造、型、软、件、包、第、版、2.0和1988．2.20。汉字造型以左上角为造型起点，字符造型以左下角为造型起点。

③ 动画轨迹坐标的确定

因为动作过程比较简单，只要知道各造型移动的最终位

置坐标，由此确定每个造型的循环体，很方便地达到过程描述的要求，各造型的最终坐标由图形数据交互生成工具GP 1407生成图1.22。相应的数据见GP 1512程序DATA语句。

④ 程序设计

由演示程序的过程描述及要求，针对已经建立的图表及图1.22各造型的位置坐标，完成程序设计。

⑤ 源程序

```
5   REM   GP1512
10  DATA   1,70,50,3,2,90,50,3,3,
     109,50,3,4,128,50,3,5,146,50
     ,3,6,166,50,3,7,186,50,3,8,1
     11,73,3,9,146,73,3,11,127,82
     ,3,10,109,102,3
20  DIM A%(20),B%(20),C%(20),D%(2
     0)
30  FOR I = 1 TO 11
40  READ A%(I),B%(I),C%(I),D%(I)
50  NEXT
60  POKE 232,0: POKE 233,96
70  SCALE= 1: ROT= 0
80  HGR : POKE  - 16302,0
90  HGR2
100  FOR I = 1 TO 7
110  FOR J = 1 TO 7
120  ROT= 7 * J: GOSUB 430
130  NEXT J
140  HCOLOR= 3: ROT= 0: DRAW A%(I
     ) AT B%(I),C%(I)
150  NEXT
160  POKE  - 16300,0: POKE 230,64
```

```
170  GOSUB 390
180  POKE  - 16299,0: POKE 230,32

190  GOSUB 390
200 S = 4
210  FOR I = 0 TO 50 STEP S
220  POKE  - 16300,0: POKE 230,64

230  HCOLOR= 0: GOSUB 310
240  HCOLOR= 3: GOSUB 350
250  GOSUB 490
260  POKE  - 16299,0: POKE 230,32

270  HCOLOR= 0: GOSUB 310
280  HCOLOR= 3: GOSUB 350
290  NEXT I
300  GOSUB 510
305  END
310  FOR J = 8 TO 11
320  DRAW A%(J) AT B%(J),C%(J) +
     50 - I
330  NEXT
340  RETURN
350  FOR J = 8 TO 11
360  DRAW A%(J) AT B%(J),C%(J) +
     50 - I - S
370  NEXT
380  RETURN
390  FOR I = 1 TO 7
400  DRAW A%(I) AT B%(I),C%(I)
410  NEXT
420  RETURN
430 X = 60 *  RND (1):Y = 160 *  RND
     (1)
440  HCOLOR= 3: DRAW A%(I) AT X,Y
```

```
450  GOSUB 490
460  HCOLOR= 0: DRAW A%(I) AT X,Y

470  RETURN
480  DRAW A%(I) AT B%(I),C%(I) +
     50
490 P =  PEEK (49152): IF P > 128
      THEN  HOME : TEXT : GET A$:
      END
500  POKE 49168,0: RETURN
510  HCOLOR= 3: POKE  - 16300,0
515  FOR I = 1 TO 40 STEP 2
520  HPLOT 67 - I,40 - I TO 202 +
     I,40 - I TO 202 + I,110 + I TO
     67 - I,110 + I TO 67 - I,40 -
     I
530  NEXT
540  RETURN
```

⑥ 程序说明

5—305句为主程序，10—50句将图1.22各造型位置坐标存入数组A%(I),B%(I),C%(I),D%(I)中，100—150句为将通、用、造、型、软、件和包随机旋转动态组合一行"通用造型软件包",160—190句为分别在第二页和第一页画出此行。200—290句为将1988. 2.20. 第2.0版由下向上移动到屏幕的适合位置，300句为将矩形由小到大,并留下相应的矩形。

310—340和350—380子程序分别在不同位置画或抹"1988. 2.20. 第2.0版",以达到换页显示的目的。

390—420句为画"通用造型软件包"的子程序。

490—500子程序为键盘接收及控制语句，当按任意键程

序就结束，如果将END改为PRINT CHR$(4)；"RUN MENL"，则按任意键可运行名为MENU的程序。

510—540子程序为画一系列由小到大的矩形。

上面设计的程序为一个通用的演示程序，只要用相应的造型替换掉图1.22中的各个造型，就能得到同样的演示效果，比如用 3.0替换第11个造型 2.0，用1989.8替换第10个造型 1988. 2.20，则由下向上移动"1989.8.第3.0版"。

远行GP 1512需要图1.22汉字及字符的造型表。

四、需要变化的动态图形设计

在计算机定性模拟中，常用变化的图形来反映运动的变化过程，如一个杠杆式物体在受力状态下变形，机构动态图形或者小孩子走路。用高级语言设计这一类动态图形一般有三种方法，其一、用节点法来实现，即所有变化的动态图形全用 HPLOT 语句编写相应的程序。主要适用于图形简单、且容易用函数关系来表达的变化图形；其二、用造型法来实现，即制备动画过程中的一系列造型，用软件来控制不同时刻显示不同的造型，使人感觉图形在变化；其三、用点线法与造型法组合来实现，即把那些计算量小，容易用函数关系来表达且又要求变化的图形用点线法来实现，而把那些不规则，计算量大的且要求不变的图形用造型来实现，这种方法适用于一般情况下变化的动态图形。关于图形和造型表生成可用前面几章介绍的一系列工具程序。下面我们用简单几个实例来说明这类动态图形的制作。

1. 旋转，逐渐变大的小旗的移动

小旗的计算量比较简单，且图形又不复杂，完全可以用节点法来实现。

① 源程序（见GP 1513）

```
5  REM    GP1513
10  HGR2 :Y = 80
20  FOR X = 30 TO 240 STEP 10
25 ST = 30 - X:A = 25 + X / 15:B =
     10 + X / 15
30 C = 3: GOSUB 60
35  FOR J = 1 TO 100: NEXT
40 C = 0: GOSUB 60
45  NEXT X
50 C = 3: GOSUB 60
55  END
60 T = 3.14159 / 180 * ST
70 A1 = A *  COS (T) + 0 *  SIN (T)
     :B1 = 0 *  COS (T) - A *  SIN
     (T)
75 A2 = A *  COS (T) + B *  SIN (T)
     :B2 = B *  COS (T) - A *  SIN
     (T)
80 A3 = 0 *  COS (T) + B *  SIN (T)
     :B3 = B *  COS (T) - 0 *  SIN
     (T)
85 A4 = 0 *  COS (T) + (B + A / 2) *
      SIN (T):B4 = (B + A / 2) *  COS
     (T) - 0 *  SIN (T)
90  HCOLOR= C
95  HPLOT X,Y TO X + A1,Y + B1
100  HPLOT X + A1,Y + B1 TO X + A2,
     Y + B2
105  HPLOT X + A2,Y + B2 TO X + A3,
     Y + B3
110  HPLOT X,Y TO X + A4,Y + B4
115  RETURN
```

② 程序说明

5—55句为主程序，25句是小旗变换的角度，实际放大的边长，这些都是循环变量X的函数，移动画面由20—45句的循环实现。

60—115句子程序是实现放转变换及画小旗的模块，旋块变换是相对于参考点(X，Y)进行的。

2. 模拟人的跑步运动

人的跑步运动，实际上是一个变化的图形动画，用节点法与造型法组合来制备3幅不变的图形，在不同时刻显示不同的图形，见图1.23。人的头部用造型，身子用HPLOT语句实现。

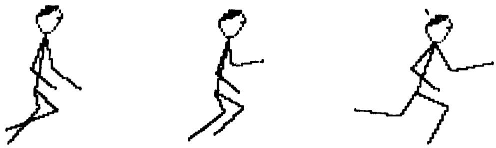


图 1.23 跑步运动不同时刻显示的图形

① 源程序（见GP 1514）

```
5   REM     GP1514
6   DATA  1,0,4,0,45,12,12,100,100,10
      0,63,79,33,228,228,60,63,63,46,
      45,53,63,63,63,46,45,61,62,63,5
      5,61,55,54,54,49,14,118,94,0
8   FOR I = 1 TO 39
9   READ A: POKE 24575 + I,A: NEXT
15   HGR2 :Y = 100: SCALE= 1: ROT= 0
```

```
20 K1 = 232:K2 = 233: HCOLOR= 3
25  HPLOT 0,160 TO 270,160
30  POKE K1,0: POKE K2,96
40  FOR X = 40 TO 240
65 C = 3: GOSUB 300:C = 0: GOSUB 300

75 X = X + 4
80 C = 3: GOSUB 500:C = 0: GOSUB 500

90 X = X + 4
95 C = 3: GOSUB 360:C = 0: GOSUB 360
110  NEXT
120 C = 3: GOSUB 300
200  END
300  HCOLOR= C: DRAW 1 AT X,Y
305  HPLOT X,Y TO X + 4,Y + 20 TO X +
     16,Y + 30
308  HPLOT X,Y TO X - 4,Y + 30
310  HPLOT X - 4,Y + 30 TO X + 6,Y +
     42 TO X - 16,Y + 53
320  HPLOT X - 4,Y + 30 TO X - 2,Y +
     40 TO X - 15,Y + 60
340  HPLOT X,Y TO X - 6,Y + 20 TO X +
     5,Y + 30
350  RETURN
360  HCOLOR= C: DRAW 1 AT X,Y
365  HPLOT X,Y TO X + 7,Y + 15 TO X +
     25,Y + 10
370  HPLOT X,Y TO X - 8,Y + 28
380  HPLOT X - 8,Y + 28 TO X + 6,Y +
     35 TO X - 2,Y + 55
390  HPLOT X - 8,Y + 28 TO X - 14,Y +
     40 TO X - 34,Y + 37
```

```
400   HPLOT X,Y TO X - 10,Y + 15 TO X
       + 2,Y + 26
420   RETURN
500   HCOLOR= C: DRAW 1 AT X,Y
510   HPLOT X,Y TO X + 2,Y + 15 TO X +
      15,Y + 12
520   HPLOT X,Y TO X   5,Y + 30
530   HPLOT X - 5,Y + 30 TO X + 7,Y +
      38 TO X - 5,Y + 53
540   HPLOT X - 5,Y + 30 TO X - 1,Y +
      40 TO X - 16,Y + 50
550   HPLOT X,Y TO X - 5,Y + 20 TO X +
      5,Y + 30
560   RETURN
```

② 程序说明

5—200句为主程序，8—9句将人头部造型表存入＄6000为起始处，25句为画地面；40—110句为模拟人的跑步运动。

300—350为画图1.23(a)子程序。

360—420为画图1.23(b)子程序。

500—560为画图1.23(c)子程序。

五、轴的结构及其装配过程的模拟

图1.24(a)、(b)为两种轴的结构及其轴系部件，现要求显示这两种轴的结构并模拟轴系零件的装配过程。

1. 过程描述

图1.24(a)轴的结构，其轴上零部件装配顺序为：圆柱齿轮、长轴套和右端轴承从轴的右端装入，左轴承、短轴套、轴承透盖和半联轴器则从轴的左端依次装入。图1.24(b)轴的结构，其轴上零部件装配顺序为：圆柱齿轮、阶梯轴套、左端轴承和半联轴器依次由轴的左端装入；仅有右端轴承从

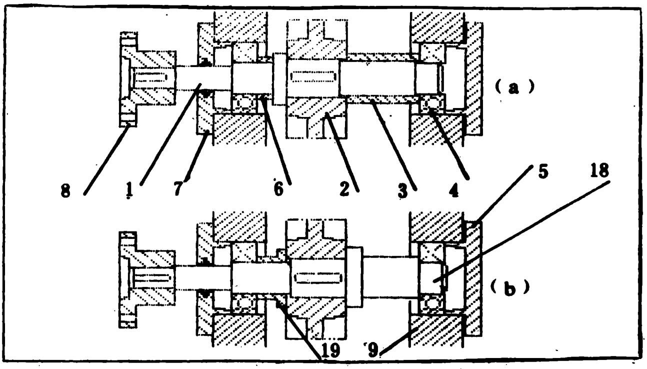


图 1.24 轴的两种结构形式

1.a轴；2.齿轮；3.右端长轴套；4.轴承；5.右端轴承盖；6.左端短轴套；7.左端透盖；8.半联轴器；9.箱体；18.b轴；19.左端阶段轴套。

轴的右端装入。从这两种的装配过程可知，前者较后者多增加了一个作为轴向定位的长轴套。

2．图形制作

图1.24全部用造型，由前述工具完成，模拟装配过程共有32个造型。第一个造型代表图1.24(a)轴，第二个造型代表齿轮，第三个造型代表右端长轴套，第六个造型代表左端短轴套，第七个造型代表左端轴承透盖，第八个造型代表半联轴器，第九个造型代表箱体，第十一第十七与第二十一第二十一个造型代表补线线条，第十八个造型代表图(b)轴，第十九个造型代表左端阶梯轴套，第二十二一第二十三个造

型分别代表(a)与(b)造型，第二十四—第三十二为1—9的数字字符造型。

3．动画轨迹坐标的确定

由图形数据交互生成工具GP1407生成相应的图形以及关键过程的坐标，所谓关键过程即零件在装配过程发生改变的一瞬间，找出这一瞬间的坐标，作为条件语句判定的依据。

4．关键问题的处理

在轴系零部件的装配过程中，比如圆柱齿轮的装配过程，在与轴相互重叠时，要抹去被轴遮去的线条，而且由于轴的直径不同，其抹去的线条长短不一样，见图1.25(a)(b)，实现的方法是利用补线的原理，采用多个造型在不同的位置显示相应多个造型来实现，图1.25(b)由3个造型组成，图1.25(d)由两个造型组成。

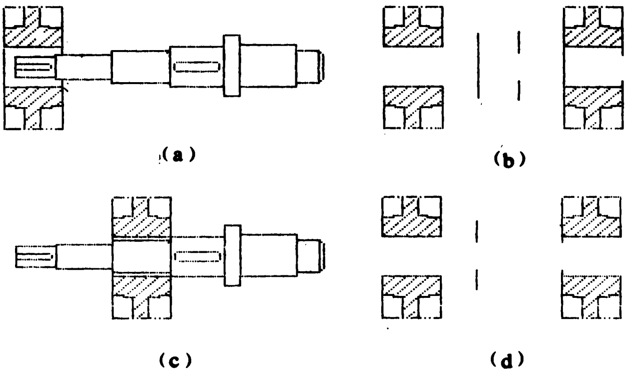


图 1.25　齿轮装配的补线原理

① 源程序（见GP1515）

```
5  REM   GP1515
9  SCALE= 1: ROT= 0
10  POKE 232,0: POKE 233,96: HCOLOR=
     3:K = 0
30  HGR : POKE  - 16302,0
32  HGR2   POKE 230,64
36  GOSUB 950: POKE 230,32: GOSUB 95
     0
38  FOR I = 1 TO 500: NEXT I
39 A = 0:X = 30:Y =  - 13
40  POKE 230,32: GOSUB 1000: POKE 23
     0,64: GOSUB 1000
45 A = 1:X = 30:Y = 95
50  POKE 230,32: GOSUB 1000: POKE 23
     0,64: GOSUB 1000
67  FOR I = 1 TO 1000: NEXT I
68 A = 0:X = 30:Y =  - 13: GOSUB 700

70  FOR I = 1 TO 500: NEXT I
71 K = K + 1: GOTO 108
72 A = 1:X = 30:Y =  - 13: GOSUB 800

74  FOR I = 39 TO 121 STEP 2
75  POKE  - 16300,0: POKE 230,64
80  HCOLOR= 0: GOSUB 2000: HCOLOR= 3
     : GOSUB 2100
90  POKE  - 16299,0: POKE 230,32
95  HCOLOR= 0: GOSUB 2000: HCOLOR= 3
     : GOSUB 2100
102  GOSUB 2300: NEXT I: GOSUB 2350
108  IF A = 1 THEN 321
110  FOR I = 182 TO 126 STEP  - 2
115  POKE  - 16300,0: POKE 230,64
120  HCOLOR= 0: GOSUB 1900: HCOLOR=
     3: GOSUB 1910
```

```
130  POKE  - 16299,0: POKE 230,32
135  HCOLOR= 0: GOSUB 1900: HCOLOR=
     3: GOSUB 1910
142  GOSUB 2300: NEXT I: GOSUB 2350
148  GOSUB 2350
180  FOR I = 182 TO 151 STEP  - 1
190  POKE  - 16300,0: POKE 230,64
195  HCOLOR= 0: GOSUB 2200: HCOLOR=
     3: GOSUB 2220
210  POKE  - 16299,0: POKE 230,32
220  HCOLOR= 0: GOSUB 2200: HCOLOR=
     3: GOSUB 2220
228  GOSUB 2305: NEXT I: GOSUB 2350
240  HCOLOR= 3: DRAW 3 AT 150,130
250  FOR I = 50 TO 117
260  POKE  - 16300,0: POKE 230,64
270  HCOLOR= 0: GOSUB 2400: HCOLOR=
     3: GOSUB 2420
290  POKE  - 16299,0: POKE 230,32
300  HCOLOR= 0: GOSUB 2400: HCOLOR=
     3: GOSUB 2420
310  GOSUB 2310: NEXT I
321  GOSUB 2350
322  IF A = 0 THEN 340
323  FOR I = 54 TO 122
324  POKE  - 16300,0: POKE 230,64
326  HCOLOR= 0: GOSUB 2130: HCOLOR=
     3: GOSUB 2150
330  POKE  - 16299,0: POKE 230,32
332  HCOLOR= 0: GOSUB 2130: HCOLOR=
     3: GOSUB 2150
335  GOSUB 2305: NEXT I: GOSUB 2350
340  FOR I = 45 TO 100
345  POKE  - 16300,0: POKE 230,64
350  HCOLOR= 0: GOSUB 2500: HCOLOR=
     3: GOSUB 2520
360  POKE  - 16299,0: POKE 230,32
```

```
365  HCOLOR= 0: GOSUB 2500: HCOLOR=
     3: GOSUB 2520
375  IF A = 0 THEN  GOSUB 2315
378  IF A = 1 THEN  GOSUB 2310
380  NEXT I
385  GOSUB 2350
390  FOR I = 194 TO 183 STEP  - 1
400  POKE  - 16300,0: POKE 230,64
410  HCOLOR= 0: GOSUB 2600: HCOLOR=
     3: GOSUB 2620
440  POKE  - 16299,0: POKE 230,32
450  HCOLOR= 0: GOSUB 2600: HCOLOR=
     3: GOSUB 2620
475  IF A = 0 THEN  GOSUB 2320
478  IF A = 1 THEN  GOSUB 2315
480  NEXT I
481  GOSUB 2350
482  HCOLOR= 3: GOSUB 1300
485  FOR I = 63 TO 98
490  POKE  - 16300,0: POKE 230,64
495  HCOLOR= 0: GOSUB 2700: HCOLOR=
     3: GOSUB 2710
505  POKE  - 16299,0: POKE 230,32
510  HCOLOR= 0: GOSUB 2700: HCOLOR=
     3: GOSUB 2710
517  IF A = 0 THEN  GOSUB 2325
519  IF A = 1 THEN  GOSUB 2320
520  NEXT I
525  GOSUB 2350
530  FOR I = 210 TO 195 STEP  - 1
540  POKE  - 16300,0: POKE 230,64
545  HCOLOR= 0: GOSUB 2800: HCOLOR=
     3: GOSUB 2820
560  POKE  - 16299,0: POKE 230,32
565  HCOLOR= 0: GOSUB 2800: HCOLOR=
     3: GOSUB 2820
```

```
575  IF A = 0 THEN  GOSUB 2330
578  IF A = 1 THEN  GOSUB 2325
580  NEXT I
585  GOSUB 2350
590  HCOLOR= 0: POKE 230,32: DRAW 11
      AT I - 1,124
595  POKE 230,64: DRAW 11 AT I - 1,1
     24
600  FOR I = 60 TO 72
605  POKE  - 16300,0: POKE 230,64
610  HCOLOR= 0: GOSUB 2900: HCOLOR=
     3: GOSUB 2920
625  POKE  - 16299,0: POKE 230,32
630  HCOLOR= 0: GOSUB 2900: HCOLOR=
     3: GOSUB 2920
645  IF A = 0 THEN  GOSUB 2335
647  IF A = 1 THEN  GOSUB 2330
650  NEXT I
655  POKE  - 16299,0: POKE 230,64
656  IF A = 0 THEN  GOSUB 2335
657  IF A = 1 THEN  GOSUB 2330
658  IF K = 1 THEN  FOR I = 1 TO 300
     0: NEXT I:K = K + 1: GOTO 72
659  FOR I = 1 TO 3000: NEXT I
660  HOME : TEXT
664  VTAB 10: HTAB 8: INPUT "Do you
     continue?(Y/N)";A$
665  IF A$ = "N" THEN  HOME : TEXT :
      END
668  GOTO 658
690  END
700  HGR : POKE  - 16302,0: HGR2
720  POKE 230,32: GOSUB 1000
730  DRAW 1 AT 190,150
735  POKE 230,64: GOSUB 1000
740  DRAW 1 AT 190,150
```

```
750  RETURN
800  HGR : POKE  - 16302,0: HGR2
805  HCOLOR= 3
815  POKE 230,32: GOSUB 1000
820  DRAW 18 AT 191,140
830  POKE 230,64: GOSUB 1000
835  DRAW 18 AT 191,140
850  RETURN
900  POKE 230,32: DRAW 1 AT 190,150
920  POKE 230,64: DRAW 1 AT 190,150
930  RETURN
950  DRAW 1 AT 190,47: DRAW 18 AT 19
     2,145
952  DRAW 22 AT 215,47: DRAW 23 AT 2
     15,155
955  HPLOT 0,0 TO 278,0 TO 278,191 TO
     0,191 TO 0,0
960  RETURN
1000  IF A = 1 THEN  GOSUB 1240
1002  GOSUB 955
1003  IF A = 0 THEN  GOSUB 1210
1005  GOSUB 955
1010  DRAW 2 AT X + 94,41 + Y: DRAW
     4 AT X + 71,33 + Y
1040  DRAW 5 AT X + 164,69 + Y: DRAW
     7 AT X + 69,32 + Y
1070  DRAW 8 AT X + 44,45 + Y: DRAW
     9 AT X + 62,32 + Y
1090  DRAW 10 AT X + 85,34 + Y: DRAW
     4 AT X + 153,33 + Y
1110  DRAW 9 AT X + 148,32 + Y: DRAW
     12 AT X + 25,Y + 43
1115  DRAW 10 AT 148 + X,Y + 34
1117  RETURN
1210  DRAW 1 AT X + 160,60 + Y: DRAW
     6 AT X + 88,43 + Y
```

```
1220  DRAW 3 AT X + 120,40 + Y
1228  HCOLOR= 0: DRAW 23 AT 215,47
1229  HCOLOR= 3
1230  DRAW 22 AT 215,47: RETURN
1240  DRAW 18 AT X + 162,50 + Y
1245  DRAW 19 AT X + 94,41 + Y
1255  HCOLOR= 0: DRAW 23 AT 215,47
1257  HCOLOR= 3
1270  DRAW 22 AT 215,47: RETURN
1300  POKE  - 16299,0: POKE 230,32
1305  GOSUB 1320
1310  POKE  - 16300,0: POKE 230,64
1315  GOSUB 1320
1320  DRAW 9 AT 92,122: DRAW 10 AT 1
     15,124
1330  DRAW 9 AT 178,122: DRAW 10 AT
     178,124
1340  DRAW 16 AT 92,118: DRAW 15 AT
     201,122
1380  RETURN
1900  DRAW 2 AT I,131
1902  IF I > 166 THEN  DRAW 11 AT I +
     26,132
1904  RETURN
1910  DRAW 2 AT I   2,131
1912  IF I > 168 THEN  DRAW 11 AT I +
     24,125
1920  RETURN
2000  DRAW 2 AT I,131
2002  IF I < 99 THEN  DRAW 13 AT I,1
     30
2004  IF I < 73 THEN  DRAW 13 AT I +
     26,130
2006  IF I > 73 AND I < 99 THEN S1 =
     11: DRAW 21 AT I + 26,127
2008  IF I > 73 THEN  DRAW 21 AT I,1
     27
```

```
2010  IF I < 56 THEN  DRAW 11 AT I,1
      25
2020  RETURN
2100  DRAW 2 AT I + 2,131
2102  IF I < 97 THEN  DRAW 13 AT I +
      2,130
2104  IF I < 71 THEN  DRAW 13 AT I +
      28,130
2106  IF I > 71 AND I < 97 THEN S1 =
      12: DRAW 21 AT I + 28,127
2108  IF I > 73 THEN  DRAW 21 AT I +
      2,127
2115  IF I < 54 THEN  DRAW 11 AT I +
      2,125
2120  RETURN
2130  DRAW 19 AT I,131
2132  IF I > 97 THEN  DRAW 18 AT 191
      ,140
2134  IF I > 53 AND I < 100 THEN S1 =
      13: DRAW 13 AT I,130
2136  IF I > 67 AND I < 112 THEN S1 =
      14: DRAW 13 AT I - 13,130
2138  IF I < 69 THEN  DRAW 12 AT I -
      13,132
2140  RETURN
2150  DRAW 19 AT I + 1,131
2152  IF I > 96 THEN  DRAW 18 AT 191
      ,140
2154  IF I > 52 AND I < 99 THEN S1 =
      13: DRAW 13 AT I + 1,130
2156  IF I > 66 AND I < 111 THEN S1 =
      14: DRAW 13 AT I - 12,130
2158  IF I < 68 THEN  DRAW 12 AT I -
      12,132
```

```
2180  RETURN
2200  DRAW 3 AT I,130
2205  IF I > 160 THEN  DRAW 12 AT I +
     32,132
2220  DRAW 3 AT I   1,130
2225  IF I > 161 THEN  DRAW 12 AT I +
     31,132
2230  RETURN
2300  DRAW 24 AT 20,100: RETURN
2305  DRAW 25 AT 20,100: RETURN
2310  DRAW 26 AT 20,100: RETURN
2315  DRAW 27 AT 20,100: RETURN
2320  DRAW 28 AT 20,100: RETURN
2325  DRAW 29 AT 20,100: RETURN
2330  DRAW 30 AT 20,100: RETURN
2335  DRAW 31 AT 20,100: RETURN
2340  DRAW 32 AT 20,100: RETURN
2350  HCOLOR= 0: POKE 230,32: DRAW 3
     4 AT 20,100
2360  RETURN
2400  DRAW 6 AT I,133
2401  IF I > 106 THEN  DRAW 1 AT 190
     ,150
2402  IF I < 64 THEN  DRAW 12 AT I -
     7,132
2404  IF I < 57 THEN  DRAW 12 AT I,1
     32
2406  IF I < 106 AND I > 57 THEN S1 =
     15: DRAW 13 AT I - 7,130
2408  IF I < 99 AND I > 55 THEN S1 =
     14: DRAW 13 AT I,130
2418  RETURN
2420  DRAW 6 AT I + 1,133
2422  IF I > 106 THEN  DRAW 1 AT 190
     ,150
2425  IF I < 63 THEN  DRAW 12 AT I
     6,132
```

```
2430  IF I < 56 THEN  DRAW 12 AT I +
     1,132
2440  IF I < 105 AND I > 57 THEN S1 =
     16: DRAW 13 AT I - 6,130
2450  IF I < 98 AND I > 55 THEN S1 =
     15: DRAW 13 AT I + 1,130
2490  RETURN
2500  DRAW 4 AT I,123
2502  IF I < 57 THEN  DRAW 12 AT I -
     1,132
2504  IF I < 46 THEN  DRAW 12 AT I +
     10,132
2506  IF I < 100 AND I > 42 THHN S1
     17: DRAW 13 AT I - 1,130
2508  IF I < 89 AND I > 45 THEN S1 =
     16: DRAW 13 AT I + 10,130
2518  RETURN
2520  DRAW 4 AT I + 1,123
2525  IF I < 56 THEN  DRAW 12 AT I,1
     32
2530  IF I < 45 THEN  DRAW 12 AT I +
     11,132
2540  IF I < 99 AND I > 41 THEN S1 =
     17: DRAW 13 AT I,130
2550  IF I < 88 AND I > 45 THEN S1 =
     17: DRAW 13 AT I + 11,130
2580  RETURN
2600  DRAW 4 AT I,123
2602  DRAW 12 AT I + 10,132
2610  RETURN
2620  DRAW 4 AT I   1,I23
2625  DRAW 12 AT I + 9,132
2630  RETURN
2700  DRAW 7 AT I,122
2701  IF I < 70 THEN  DRAW 12 AT I
     14,132
```

```
2702  IF I > 69 AND I < 89 THEN S1 =
     18: DRAW 14 AT I - 14,130
2703  IF I < 81 THEN  DRAW 14 AT I -
     7,130
2704  DRAW 17 AT I + 1,123
2708  RETURN
2710  DRAW 7 AT I + 1,122
2711  IF I < 69 THEN  DRAW 12 AT I -
     13,132
2712  IF I > 70 AND I < 88 THEN S1 =
     19: DRAW 14 AT I - 13,130
2713  IF I < 80 THEN  DRAW 14 AT I -
     6,130
2715  DRAW 17 AT I + 2,123
2718  RETURN
2720  DRAW 7
2730  DRAW 10 AT 148 + K1,34 + Y1
2735  RETURN
2800  DRAW 5 AT I,159: DRAW 11 AT I
     1,124
2818  RETURN
2820  DRAW 5 AT I - 1,159: DRAW 11 AT
     I + 2,124
2830  RETURN
2900  DRAW 8 AT I,135: DRAW 12 AT I
     19,132
2910  RETURN
2920  DRAW 8 AT I + 1,135: DRAW 12 AT
     I - 18,132
2930  RETURN
```

② 程序说明

5—690句为主程序，10—38句在两页上画出两根轴并停一会，39—67句擦掉轴并画图，然后停一会。

68句为显示图1.25(a)及其相应轴，见图1.26（a），110—142句为模拟圆柱齿轮从轴的右端装入，180—228句为模拟长轴套从轴的右端装入，250—310句为模拟短轴套从轴的左端装入，340—380句为模拟左端轴承从轴的左端装入，390—480句为模拟右端轴承从轴的右端装入。482句为画出箱

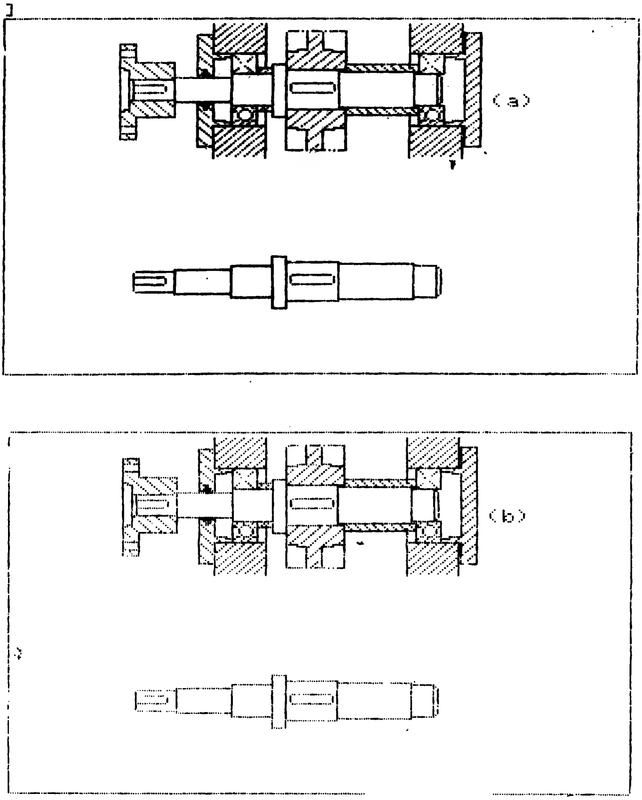


图 1.26 两种轴的结构

体，485—520句为模拟左端透盖从轴的左端装入，530—580句为模拟轴承盖从轴的右端装入，600—650句为模拟半联轴器从轴的左端装入。

72句为显示图1.25(b)及其相应轴，见图1.26(b) 74—102句为模拟圆柱齿轮从轴的左端装入，323—325句为模拟阶梯轴套从轴的左端装入。左轴承、右轴承、箱体、透盖、轴承盖和联轴器的装配见上述。

700—750句为画图1.26(a)子程序。

800—850句为画图1.26(b)子程序。

950—960句为画两根轴子程序。

1000—1315句为画图1.25子程序。

运行程序GP1515需要图1.24零件的造型表。

第三节　从数据结构来生成一幅动画

前面介绍了中华学习机图形显示原理，动态图形设计技巧并给出了一些实例程序，基本上都是首先构造动画的模式，然后用程序来实现，下面试图从数据结构的角度来生成一幅由造型法来实现的动画，即改变数据和相应的造型，就能得到满足预定要求的动态图形，改变数据和造型借用图形数据交互生成工具与造型生成工具来完成。

一、数据结构

前述几章介绍了从索引变形树数据结构来生成一幅图形，并给出了相应的程序操作，存贮各图形的指针的表可以看为索引表，从动画的角度考虑，可以增加辅助索引，存放各图形索引的指针以及关键字，其相应的数据结构见图1.27。

B%(I)数组存贮每幅子图形的图形数据，实际上是各

造型移动的轨迹坐标以及颜色；A%(I)数组按顺序存贮各图形数据的首指针及尾指针，以便于各图形的正方向与反方

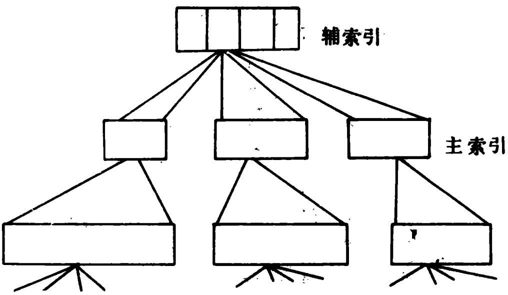


图 1.27 二级索引的树结构

向移动的实现，C%(I)数组存贮主索引表指针及其动态图形的关键字，比如该关键字用来表示：双页动画、单页动画，在某处显示提示信息或图形、声音等等，有关该种结构的交互实现，程序稍大也较复杂，由于本书的篇幅，这里就从略，下面用个简单的动画实例来说明该种结构的程序应用。

二、程序实现

① 用数组来实现索引结构

为了方便我们用两个造型“中”和“国”组成的屏幕，

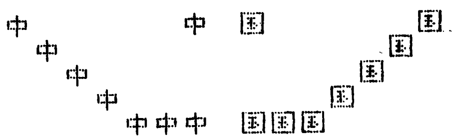


图 1.28 汉字屏幕

见图1.28，并要求它们按其路径同时移动，看看如何由数据，结构来生成相应的动画。

相应的路径坐标见表1.4。

表 1.4

Drawing = 1

1	100	60	3	1	40	60	3
1	50	70	3	1	60	80	3
1	70	90	3	1	80	100	3
1	90	100	3	1	100	100	3
2	120	58	3	2	180	58	3
2	170	68	3	2	160	78	3
2	150	88	3	2	140	98	3
2	130	98	3	2	120	98	3

上述数据有两个造型，分别调用 8 次，除去各造型的颜色，共有64个数据。我们设置C%(Ⅰ)、A%(Ⅰ)、B%(Ⅰ)三个数组按照图1.27索引结构来组织这些数据，从而实现相应的动画。

B%(Ⅰ)数组存图形数据，A%(Ⅰ)安排主索引表，C%(Ⅰ)数组安排辅索引表。

B%(0)存放每个造型被调用的次序，即8次；B%(1)—B%（64）存放图形数据；A%(1)—A%(2)存放第一个造型在B%(Ⅰ)数组中的位置，即指针 1 和32，A%(5)—A%(6)存贮第 2 个造型在A%(Ⅰ)数组中的位置，即指针33—64；C%(0)存放同时动画的造型个数，即2，C%(1)—C%(2)存贮第二个被调用造型图形数据指针在A%(Ⅰ)数组中的位置，即指针5—6，C%(6)—C%(7)存贮第一个被调用造型图形数据指针在A%(Ⅰ)数组中的位置，即1—2。

② 源程序（见GP1516）

```
5  REM   GP1516
8  DIM B%(100),A%(20),C%(20)
10  DATA  1,100,60,3,1,40,60,3,1,50
    ,70,3,1,60,80,3
30  DATA  1,70,90,3,1,80,100,3,1,90
    ,100,3,1,100,100,3
50  DATA  2,120,58,3,2,180,58,3,2,1
    70,68,3,2,160,78,3
70  DATA  2,150,88,3,2,140,98,3,2,1
    30,98,3,2,120,98,3
80  DATA 2,0,14,0,32,0,0,0,100,44,4
    5,45,45,45,81,41,45,45,53,54,63
    ,63,63,36,76,33,52,54,54,54,54,
    0
90  DATA 128,81,10,45,45,45,54,54,5
    4,54,63,63,63,39,36,36,36,164,7
    4,42,237,51,62,45,221,54,47,45,
    9,220,148,0
95  FOR I = 1 TO 64: READ B%(I): NEXT
100  FOR I = 1 TO 64: READ A: POKE 2
    4575 + I,A
105  NEXT
110 B%(0) = 8
120 A%(1) = 1:A%(2) = 32
130 A%(5) = 33:A%(6) = 64
140 C%(1) = 5:C%(2) = 6:C%(3) = 0:C%
    (4) = 0
145 C%(0) = 2
150 C%(6) = 1:C%(7) = 2:C%(8) = 0
155  SCALE= 1: ROT= 0:K1 = 232:K2 =
    233
160  HGR2 : HCOLOR= 3: POKE K1,0: POKE
    K2,96
170  FOR L = 0 TO B%(0) - 1
```

```
175  FOR J = 1 TO C%(0)
180 I0 = C%(1 + 5 * (J - 1))
200  HCOLOR= 3:K = A%(I0): GOSUB 400

210  NEXT J
215  FOR J2 = 1 TO 200: NEXT
220  FOR J = 1 TO C%(0)
230 I0 = C%(1 + 5 * (J - 1))
240  HCOLOR= 0:K = A%(I0); GOSUB 400

250  NEXT J
260  NEXT L
265 L = L - 1
270  FOR J = 1 TO C%(0)
275 I0 = C%(1 + 5 * (J - 1))
280  HCOLOR= 3:K = A%(I0): GOSUB 400

285  NEXT J
300  END
400 I = B%(K + 4 * L):X = B%(K + 4 *
     L + 1):Y = B%(K + 4 * L + 2)
410  DRAW I AT X,Y
420  RETURN
```

③ 程序说明

5—300句为主程序，10—70句为图形数据，由95读进B%(I)数组中，80—90句为“中”和“国”的造型数据，由100句放入$6000为起始处，110—150由B%(I)、A%(I)、C%(I)按图1.27索引结构思想安排相应的数据结构。170—210的二重循环实现“中”和“国”两个造型按照图1.27的路径由上向下移动。

400—420句为调用相应造型子程序。

如果用下面的语句替换GP1516程序相应的语句，就会实现“中”和“国”造型按图1.27由下向上移动。

```
140 C%(1) = 5:C%(2) = 6:C%(3) = 0:C%
     (4) = 0
145 C%(0) = 2
150 C%(6) = 1:C%(7) = 2:C%(8) = 0
```

如果改变移动路径坐标或用其它造型替换“中”与“国”造型，就能获得另一幅动画图形。

对GP1516程序结构，只要改其数据，就可以实现多个造型的同时移动，如速度允许的话，可以实现任意多个造型的移动。

由上面的实例可以看出，图1.27描述的结构可实现较为复杂的动画。在中华学习机实现的减速器拆装模拟系统，在PC机上实现的零件车削的仿真软件是借用该种结构思想完成的复杂动态图形模拟。

第六章　通用造型软件包2.0版

通用造型软件包是在中华学习机及其兼容微机上开发的，实现交互式键盘命令操作，具有显示与生成图形速度快，进入与退出各功能模块实行区域命令操作，操作起来十分方便，基本功能齐全，使用灵活，经济实用等特点，适用于教学软件的开发，动画以及图形库建立等。下面介绍本软件的结构，主要功能、软件实现思想及使用方法。

第一节　软件结构与功能

一、软件结构

本软件采用树型层次结构，用图形汉字提示，其结构模块图见图1.29，图中只列出了建立图表的分支模块，其它功能模块请参阅本章第三节本软件的使用方法。

二、主要功能

由图1.29可知，本软件共有6个部分组成，其中建立图表、积木块、西文编辑均能完成图表的建立，图形编辑主要用于图形库建立及动画轨迹数据的生成；实用文件包括一些实用工具。

1．造型表的生成

常见的一些图形一般包括具有一定函数关系组成的曲线。不规则的造型，包含字符图形或是复杂的数学式子造型，生成这类造型工具软件主要由建立图表、积木块、西文编辑来

实现，建立图表主要是生成具有一定函数关系的曲线造型，积木块主要是快速生成造型或是不规则的造型，而字符或复杂的数学式子造型生成用西文编辑。

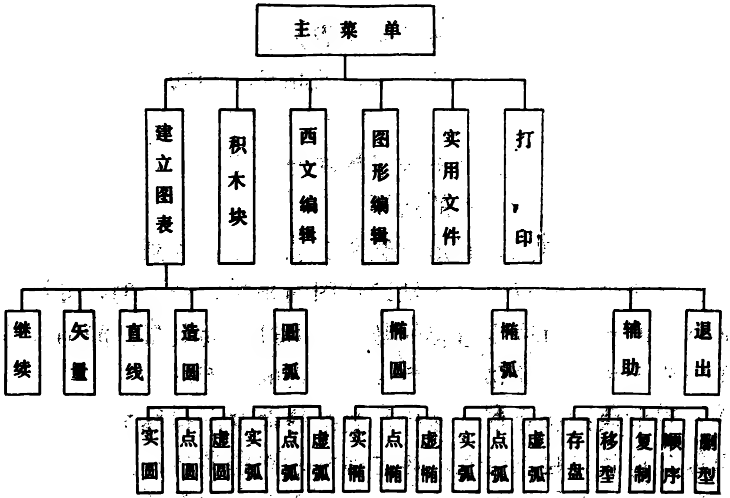


图 1.29

① 具有函数关系的造型

直线：用十字光标在屏幕上指出直线的起始点、终止点，可生成任一方向的直线。

实圆、虚圆、点划线圆：用十字光标在屏幕上指出圆心、半径，然后自动生成圆的造型。

实线椭圆、虚线椭圆、点划椭圆：用十字光标在屏幕上指出椭圆的中心、长轴、短轴，然后自动生成椭圆的造型。

实线圆弧、虚线圆弧、点划线圆弧：用十字光标在屏幕

上指出圆弧的起始点、中心及终止方向，然后自动生成圆弧造型。

实线椭圆弧、虚线椭圆弧、点划线椭圆弧：用十字光标在屏幕上指出五点：前三点代表该椭圆的形状，后两点代表椭圆的起始及终止方向。

利用指定坐标自动按参数生成造型，可十分方便地实现直线与圆及圆弧相切、任意弧段的相接或相切。

② 不规则的造型

生成这类造型主要由矢量模块来实现，或者由矢量配合积木块快速生成不规则的造型。

矢量：有网点、看形、删除及 4 个方向的画与不画的移动矢量命令。网点主要起辅助作图的坐标定位，删除可抹去屏幕上的点同时改变内存的数据，通过移动矢量，能随意地在屏幕上模拟笔的移动来生成不规则的造型。

积木块：有看形、移标、删除等命令，移标主要确定被积每个造型的起点位置，通过移标来确定各造型矢量终止及被积造型的起始位置，能方便地将多个造型合并为一个不规则的造型，实现积木式拼图，也能将多个造型表有选择地合并为一个造型表或重排造型表的次序。

③ 字符或函数式子造型

生成这类造型主要由西文编辑来实现。

西文编辑：有看形、移标、删除、西文及矢量等命令。通过西文命令可在图形状态下实现键盘字符的交互式编辑，通过移标、西文及矢量命令可生成一个复杂的函数式子造型。

2. 图形编辑

图形编辑有复制、插入、替换、换色、看形、删除等命令，能对已建立的图表中的造型实现动态的复制、插入和替换操作，也可以抹去已经复制图形的任意部分，一次可编辑多幅图形，图形数据直接存入造型表之后，占内存少。编制比较复杂的100幅图形，一次可装入RAM内，形成的数据结构相对独立。利用此项功能，可方便地制作图库、幼灯片、动画数据的生成以及各个领域的图形显示。

3. 实用文件及打印

实用文件给用户提供了一些实用工具，留有用户接口，图表压缩及图形数据格式转换。

打印主要是打印各造型的起点坐标，每幅图形的指针数据以及图形的硬拷贝。

第二节 软件实现

这里就该软件在研制过程中，介绍一些问题的处理方法及其主要功能的实现。

一、软件内存分配

主菜单上的功能模块与功能模块之间采用覆盖技术，进入某一功能模块后其运行的各变量在不同区域另开缓冲区保存，其内存分配为：

建立图表：＄800—＄3FFF之间存放浮点BASIC语言，提示信息直接放在程序之后，并与浮点BASIC一起存盘，共约14 KB内存。＄4000—＄5FFF为图形显示缓冲区，＄6000—＄9600为造型表存贮区。

积木块：＄800—＄1FFF之间存放浮点BASIC语言，＄4000—＄5FFF存放提示信息或用户子程序存贮区。

＄2000—＄3 FFF 为图形显示缓冲区。＄6000—＄9600为造型表存贮区。

西文编辑：＄800—＄1 FFF 之间存放浮点BASIC语言及提示信息，＄4000—＄5 FFF 存放键盘字符造型表，＄2000—＄3 FFF 为图形显示缓冲区。＄6000—＄9600为造型表存贮区。

图形编辑、实用文件及打印：＄800—＄3 FFF 之间存放浮点BASIC语言及提示信息，＄4000—＄5 FFF 为图形显示缓冲区，＄6000—＄9600为造型表存贮区，图形数据放在造型表之后。

二、一些问题处理

1．各功能模块变量的传递

中华学习机的造型涉及到的变量都是整型量，若这个变量小于255，对 8 位机只用一个字节就可存放，若变量大于255，则需存放两个字节中，这样做有利于提高数据传递速度与软件运行的可靠性。关于变量的存取请参阅GP1405。

2．直线线型

用HPLOT语句绘的斜线是由折线组成的，比如绘45度方向的斜线。其局部放大见图1.30。这种线型不便于作图形显示。抹去斜上方的 3 个点，即斜上方 3 个点为不画向量。用造型来实现这种线型图1.31与前 4 个点对应的十六进制数为 0 C、0 C。这样的线型特别适用于剖面线的显示。

3．连续向上移动且不画向量

按照APPLEⅡ的造型定义不能实现这一要求，因为 1 个八位数可有 3 个移动而不画向量，如果连续向上移动且不画（只要 3 个向量，也就是该造型的结束），我们是这样处理

的，如果连续向上移动 2 个不画向量，就用二进制10000000代表向上移动 1 个不画向量，实际上是向上移动 3 个不画向量同时再向下移动 1 个不画向量。

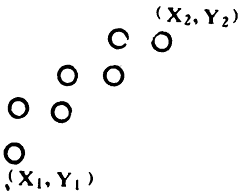


图 1.30

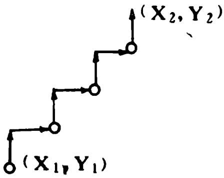


图 1.31

三、主要功能的实现

1. 按函数关系来生成造型

中华学习机的屏幕图形显示所提供的高解析度绘图有节点法和选型法，即：

① HPLOT X，Y或HPLOT X1、X1 TO X2，Y2

在屏幕上画一个点或一条线，点的坐标由（X，Y）给定，线的坐标由（X_1，Y_1），（X_2，Y_2）给定。

② DRAW K AT X，Y

在屏幕上画一个图形、图形由K定义，图形的坐标由（X，Y）给定。

其中①易于实现画圆，画直线以及其它具有函数关系的图形、存贮方式主要有两种：一种是存贮整个图形缓冲区或经压缩以后存贮缓冲区的有关区域，这属于印象存贮法；另一种是存贮图形的有关节点。而②则是用一系列的二进制编码集合来描述图形的轨迹，存贮的是整个轨迹及其移动的路

径（一系列二进制编码的集合）。按参数来生成造型，实际上包括两个方面，其一，设计交互绘制图形（用 HPLOT 语句实现）作为人与机器的交互模块；其二，需计设HPLOT所画的图形与造型生成的转换模块，其思路为：

- 交互指出关键坐标。
- 提示图形，用HPLOT X，Y实现。
- 执行转换模块。
- 显示最终图形。
- 结束。

有关转换模块的程序清单见GP 1302。

2．图形编辑及其相应的数据结构

图形编辑主要有插入、复制、删除、修改、替换。这些工作的完成主要是人与图形的交互来实现。人对图形施形操作，实际上是该图形的数据结构的操作。

该图形的数据结构采用索引文件的变形树结构，其底层结点的记录后两个字节又作为插入图形的指针，可用可不用，最终形成的索引文件及其相应的记录集合自动按紧凑格式存贮，其结构见图1.32所示。

索引文件有表头、图形关键字及其指针，其关键字即为该指针所在图形的位置。

这种结构的插入、修改、增加类似于一般高级语言所提供的随机文件，但操作均在RAM内进行，且各记录可按不等长存贮，理论上各记录的长度为2个字节所存的最大整数，这是因为记录的长度不是事先规定，而是随着图形的增加自动增加的。一个子图块的复制占4个字节。因此该种结构比随机文件要优越得多。

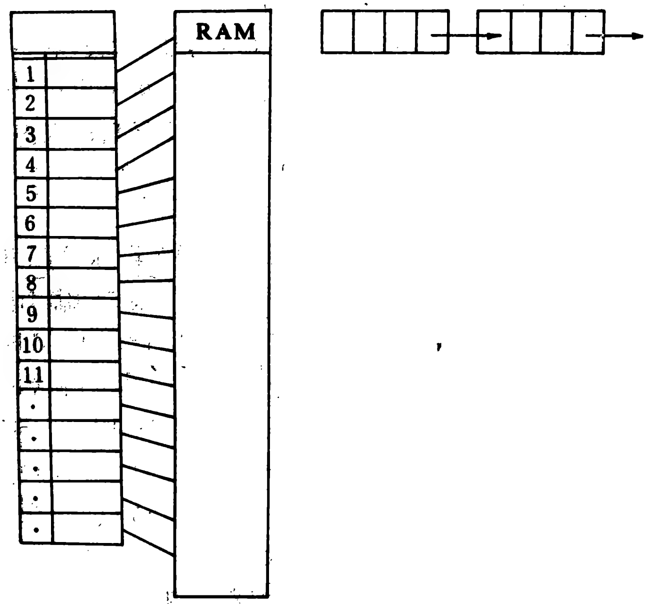


图 1.32

3. 删除

中华学习机的造型表数据的结构是实现紧凑格式排列的，各造型的结束标志十六进制为00，根据这一特点，只要用00填到造型数据的起始2个字节，就可删除相应的造型。

第三节 通用造型软件包2.0版使用说明

一、启动运行

1. 系统配置

中华学习机（或其兼容机） APPLE Ⅱ， RAM为

48KB；一个或两个磁盘驱动器，显示器和打印机。

2. 启动运行

首先将造型软件包系统盘插入1号驱动器，顺序打开主机电源及显示器，待操作系统运行完毕后，屏幕出现HELLO的动画演示，这时只要敲回车键，屏幕上随后出现图1.33。



图 1.33 主菜单屏幕

基本建立图表，西文编辑和积木块均可生成造型。建立图表主要提供了交互按参数在屏幕上指出关键点后自动生成具有一定函数关系的图块（直线、圆、椭圆、椭弧以及各种线型的弧段）；西文编辑主要是进行带图形西文的编辑和一些复杂的数学公式的建立，积木块可以让用户建立子图库，实现积木式拼积造型；图形编辑是让用户对建立的造型实行动态的复制、增加、替换，可方便地制作图库，动画（只要给出循环的起始值及终值）；实用文件提供了一些实用文件——压缩图表，格式转换及整幅图形的移动两幅图形，可打印各造型的起点坐标以及动画的起始值、终值等。

对于主菜单（见图1.33）；按空格键或“M”键，矩形方

框顺序下移，按“I”键矩形方框上移，按回车键后进入矩形框的某一功能模块，按其它键系统不反应，各功能模块互相独立，但又可通过主菜单互相联系，结束运行后返回到浮点BASIC语言状态，但此时的图表数据文件还在内存，对于只有一个驱动器的APPLEⅡ或中华学习机，可取出软件包的系统盘，插入用户盘将内存图表数据存盘。

二、建立图表

从主菜单进入该功能模块后屏幕显示：“是开始建立图表吗？(N)”，这时按回车键或其它键，表示从第一个造型开始建立图表并出现图1.34，让你交互地通过键 I，J，M或L

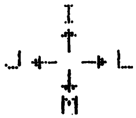


图 1.34 移动标记

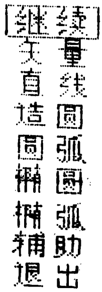


图 1.35 建立图表菜单

确定该造型的，起始点坐标。待光标位置确定好以后按“Q”键退出回到该模块的菜单见图1.35；如内存有图表式调入外存图表，继续增加造型，则进入该功能模块后，请按“N”键。

1. 继续

从主菜单进入该模块后屏幕显示：“继续吗？(N)”按N键，

表示取消该造型的编号，然后返回主菜单命令（图1.35），按回车键及其它键后，退到图1.34，让你确定该造型的起始位置；按“Q”键退到菜单命令（图1.35）。

2．矢量

进入该模块，屏幕右侧显示如下的命令：图1.36所示。只要敲上相应的字符，就可实现4个方向的画量和不画矢量的移动。

按B键即由后向前逐点地删除造型。

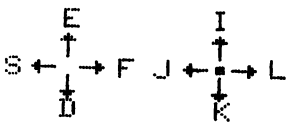


图 1.36 矢量移动标记

敲G键屏幕上出现间格为10的网点。敲A键将所有造型快速显示在屏幕上。敲Q键返回到菜单命令（图1.35）。

3．直线

进入该模块，先要按回车键输入直线的起点；这时只要按I、J、K或L键（按GRL－F慢速移动，按一次F键后进行累加，连续地按F可快速的移动）将光点移到适当的位置后，按回车键输入第二点，可看到直线的生成。通过该模块能生成任意方向的直线或多个线段组成的造型。

4．造圆

进入该模块后屏幕出现图1.37的菜单，其中实圆为实线圆，虚圆为虚线圆，点圆为点划线圆。进入实圆，虚圆或点圆模块后，只要输入两点，第一点为该圆的起点；第二点为

该圆圆周上的一点。第一点与第二点的连线需在一水平线上；然后逆时针方向生成造型，其造型的起点与终点重合后，敲Q键退出，在不退出图1.36模块的条件下，可生成同心圆的造型输入方法同前。

5. 圆弧

进入该模块后屏幕出现图1.38所示菜单，其中实弧为实线圆弧，虚弧为虚线圆弧，点弧为点划线圆弧。

实 圆
虚 圆
点 圆
退 出

图 1.37　造圆菜单

实 弧
虚 弧
点 弧
退 出

图 1.38　造弧菜单

进入实弧，虚弧或点弧模块后，只要输入 3 点，进入该模块的第一点的输入前请不要移动光标，第一点为圆弧的中心，第二点为圆弧的终止方向，图1.38所示。然后按逆时针生成造型。若想接着造另一段弧，请将该弧第一点位置与前段弧的终止位置重合（见图1.39虚线）。

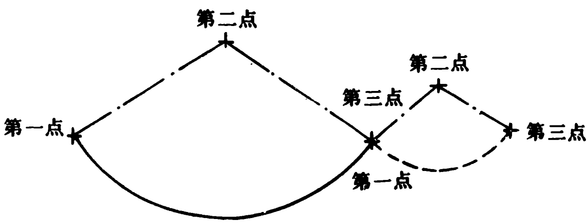


图 1.39　圆弧三点的输入

进入该模块屏幕出现图1.40，其中实椭为实线椭圆，虚椭为虚线椭圆，点椭为点划线椭圆。进入其中任一模块后，只要输入三点(进入该模块的第一点输入前请不要移动光标)，第一点为椭圆的中心，第二点与第一点的距离在模线上的投影的绝对值为该椭圆的长轴，第三点与第一点距离在纵轴上的投影的绝对值为该椭圆的短轴，见图1.41。所生成的椭圆造型起始点和终止点，均为该椭圆的中心。

6．椭弧

进入该模块后屏幕出现图1.40，其实弧为实线椭弧，虚弧为虚线椭弧，点弧为点划线椭弧，进入其中的任意模块后只要输入6点，前3点代表该椭弧所具有的椭圆形状(图1.41)，后三点的输入方法与生成圆弧时输入方法相同，如图1.39所示。

实 椭
虚 椭
点 椭
退 出

图 1.40　椭圆菜单

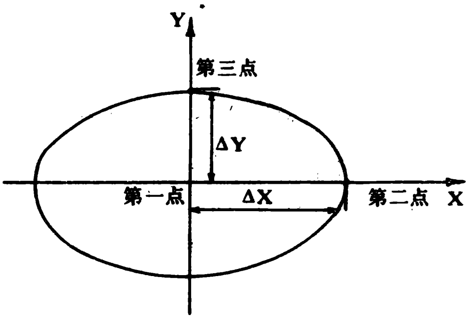


图 1.41　椭圆3点输入

7．辅助

进入该模块后屏幕出现图1.42。

① 存盘

进入该命令，屏幕显示“存盘吗?(N)”，按N键返回辅助菜单，缺省项屏幕左下角出现“Enter Name of table……:”

这时只输入相应的名字并加驱动器号即可将内存图表存入1号或2号驱动器后返回辅助菜单。

② 复制

进入该模块后屏幕出现“请输入造型”，输入相应造型编码后，屏幕即闪烁出现被选造型，且出现“是这个造型吗?(N)”，按N键则重新选择，按其它键屏幕出现图1.34，可动态地移动被选中的造型加以复制；按“E”键复制造型，按CTRL-H删除被复制的造型，该复制主要是配合造型的建立，对于同样的子图型，只要建立一个造型，其它可进行复制处理，十分方便。复制后，按“A”键即可顺序地出现所复制的图型。按“Q”键即返回到辅助菜单图1.42。

存盘
移型
复制
顺序
删型
退出

图 1.42 辅助菜单

③ 顺序

进入该模块后，按回车键或空格键就可依次看到所有的造型，按一次键出现一个造型，不看时按“Q”便返回图1.42的菜单。

④ 移型

进入该模块后屏幕出现“请输入造型”，输入相应造型编码后，屏幕上闪烁出现被选中的造型，且出现“是这个造型吗?(N)。”被选中的造型可动态地改变起始点坐标。

⑤ 删型

进入该模块后屏幕显示“请输入造型”，输入相应的造型

编码后，屏幕出现“是这个造型吗?(N)”，被选中的造型即被删除。做这项工作需十分小心，一般情况下，在文件存盘后再做这项工作。

三、西文编辑

从主菜单进入该功能模块后屏幕显示“是开始建立图表吗?(N)”，这时敲回车键或其他键表示从第一个造型开始建立图表，并出现图1.34，让你选择起始点坐标，待光标位置确定好以后按Q键退到该模块的菜单，菜单命令见图1.43，如内存有图表，请按“N”键。

继续
西文
矢量
复制
移型
删型
存盘
退出

DXR

图 1.43 西文编辑菜单

大写
小写
移标
退出

图 1.44 西文菜单

1. 继续、矢量、复制、删型、移型（同前，略）

2. 西文

进入该模块，屏幕出现图1.44。

① 大写

进入该模块，菜单消失，只要敲相应的键盘字符，即可显示该字符(大写及其它字符、数字)，敲CTRL-H可删除

该字符。按CTRL-Q可退出到图1.44。

② 小写

进入该模块，菜单消失，操作同①，这时显示的小写字母及其它字符、数字。

③ 移标

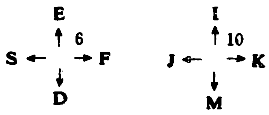


图 1.45 移标标记

进入该模块，屏幕出现图1.45，该模块的功能主要是确定通过不画矢量移动来确定造型终点的位置，其中按E、S、F、D键将移动6个不画矢量，按I，J，K，M键只移动一个不画矢量。

四、积木块

从主菜单进入该功能模块后屏幕显示，“是开始建立图表吗？(N)”，这时敲回车键或其它键表示从第一个造型开始建立图表，并出现图1.34，让你选择起始点坐标，待光标位置确定好以后，按Q键退到该模块菜单命令，见图1.46。

积木

进入该模块，屏幕的左下角出现 command: 命令,可输入以下的命令：

CATALOG：查看磁盘文件名。

BLOAD：将积木块调入内存，

MX：同“移标”命令，确定下次被积造型位置。

CDNT：继续下一个造型积木

AD：继续积木

B：由后向前逐次删除被积木的造型。

Q：退出该命令。

CATALOG、BLOAD为磁盘操作命令，在进入积木命令时，需将积木文件调入内存，其积木文件的长度为4 KB，可以在建立图表，积木块或西文编辑状态下直接生成的图表，也可以是实用文件经压缩以后存盘的图表。如积木文件超过4 KB，后面将被忽略，当超过8 KB，将冲破坏内存图表。在使用时需加以小心，具体的使用方法见后面的实例。

继　续
积　木
矢　量
复　制
移　标
删　型
移　型

图 1.46　积木块菜单

继　续
复　制
替　换
看　形
存　盘
退　出

图 1.47　图形编辑菜单

五、图形编辑

从主菜单进入该模块后，屏幕出现“是开始建立图库吗？(N)”，按回车键或其它键，屏幕出现图1.47，如不是开始建立图库，请键入N键。

1. 继续

继续下一个图形的建立，进入该模块，屏幕出现“继续吗（N）”，按回车键从下一个图形开始，按N键表示图形编码仍然处于上一个图形状态。

2．复制

进入该横块，屏幕出现“请输入造型”，输入相应的造型编码后，屏幕闪烁地出现被选中的造型，同时左下侧显示“是这个造型吗？（N）”，输入回车键，屏幕出现图1.35及被选中的造型，按“E”表示复制成功并在同一点显示一个“+”字光标，按CTRL-H可删除被复制的造型，按CTRL-F表示慢速移动，按“F”表示累加，连续按“F”键可实现快速移动，按“A”键表示看所有复制的图形。

3．替换

进入该功能模块以后，屏幕出现“请输入图形”，输入相应的图形编号并回车，屏幕显示相应的图形，同时屏幕左下角显示“是这个图形吗？（N）”。如替换造型不在显示图形中进行，请按N键，重输图形编码。否则按Q键，屏幕图形消失，这时按任意键，屏幕显示被复制的造型，如果替换其中某一造型，当显示该造型时（每按一次键只显示一个造型），请按Q键，这时屏幕显示“请输入造型，同时显示是这个造型吗？（N）”，按回车键或其它键后可将被选中的造型替换图形中相应的造型，并通过I，J，M，L 4个键可将该造型移动到相应的位置。

4．看形

进入该模块，屏幕顺序显示复制的图形，理论上图形个数可建立2个字节所存的最大整数，但由于中华学习机内存有限，复制的图形数据主要取决于图形的复杂程度，一个造

型的复制占4个字节。按Q键可退出该模块，返回菜单，见图1.47。

5. 存盘

进入该模块（操作同前，略），但可将复制图形信息以及图表一块存盘，并加后缀“·TAB”。存盘时未指明驱动器号码，自动存入2号驱动器。

6. 颜色

在图1.29状态下按“C”键，菜单消失，同时屏幕左下角出现“Enter color〈1，2，3，4，5，6，7 or 0〉：输入相应的代码，可得到不同的颜色，并返回菜单(图1.47)。

六、实用文件

进入该功能模块，如内存没有图表，屏幕左下侧显示“Enter Name shape-table:”输入相应名字后进入菜单命令，见图1.48。

用户文件
图表压缩
格式转换
退　　出

图1.48 实用文件菜单

1. 用户文件

进入该模块后，屏幕下侧显示：“Enter Name of 11ser:”，用户可以建立自己的分菜单或者建立自己的子图库的管理程序，然后以USER1，USER2，USER3这三个文件名的任何一个存入造型软件包盘中，如果用户没有建立，输入任意名字会自动返回菜单。

2. 图表压缩

进入该模块，随后屏幕左下角显示“Enter Name of C-blocks:”，输入相应的名字后，并给出相应驱动器号，则Ⅰ或Ⅱ号驱动器工作，将原有图表压缩存贮在用户盘上，进行完毕后返回到实用文件菜单，经压缩存贮的图表自动以“·C”为后缀，下次作为积木文件使用时，需在原来名字后面

加上后缀“·C”。

3．格式转换

在建立图表、西文编辑、积木块生成的造型同时也生成一幅图形，如果要保留且还要继续复制图形。请进入格式转换模块，稍后又返回到菜单命令，见图1.49，进行格式转换后再进入图形编辑时请按N键，并进入继续模块表示从第二幅图形开始。

七、打印输出

使用本功能模块之前请接通打印机电源（没有打印机可将结果打印在屏幕上）。进入该模块后如果内存有图表就直接进入菜单，否则左下角显示“Enter Name of Shape-table:”输入相应名字后进入菜单命令，见图1.49。

打印坐标
打印图形
打印指针
退　　出

图 1.49　打印清单

1．打印坐标

进入该命令后屏幕出现各个造型的起始点坐标，按Y键再回车就接通打印机，打印屏幕出现过的各造型的起始点坐标，然后返回到菜单，按其它键则不接通打印机返回到菜单，见图1.49。

2．打印图形

进入该命令，屏幕显示复制的图形，按P键接通打印机拷贝屏幕上显示的图形，按其它键，显示下个图形，按Q键直接返回到菜单命令（图1.50），对没有格式转换成没有进行图形编辑的造型表，将不会显示图形。

3．打印指针

进入该模块，可将各个图形编号及其相应的首指针打印

出来，便于动画的制作以及图形的保留。

八、简单实例

例1，已知一个造型表有下列2个造型，见图1.50（a)、(b)，试通过积木块将这两个造型合并为一个造型，见图1.50(C)（注：第一个造型的终点在中心，而第二个造型的起点在中心）。

步骤：

Ⅰ号驱动器放入系统盘，输入名字如“AA”(名字随意取)，或将图表压缩存盘。

进入积木块中的积木模块后出现：

COMMAND: <u>BLOAD↙</u>

Enter Name of C-blocks: <u>AA↙</u>

Command: <u>Cont↙</u>

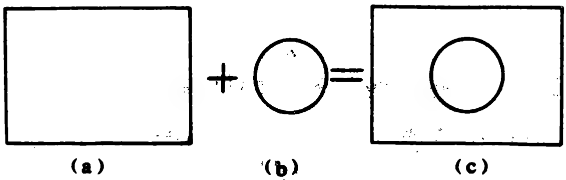


图 1.50 两个造型的拼积

请输入造型<u>1</u> ↙

是这个造型吗?(N) ↙

Command: <u>AD</u> ↙

请输入造型<u>2</u> ↙

是这个造型吗?(N) ↙

注：横线上面的字符或数字是用户输入的相应内容。

例2,已知一个造型表有任意两个造型，已按A·C存存用户盘上，现有1个造型表，其中有另外2个造型，要求将这两个造型表合并为1个造型表，并组成相应的图形。

步骤：

① 进入积木块的积木模块

command: BLOAD↙

Enter Name of C-block: A·C↙

command: cont↙

请输入造型1↙

command: CONT↙

请输入造型2↙

是这个造型吗（N）↙

command: 2↙

② 进移型或复制命令

利用移型命令将新造型表中的各造型进入交互式动态移动来组成您所需的图形。

③ 存盘

取名为AB。

例3,已知上面做好的图形，现需脱离造型软件包，在浮点BASIC状态下调用该图形

方法1

① 立即方式

```
] BLOAD  AB, D2
] BRLIN  BTABLE, D1
```

② 间接方式

```
10  A$ =CHR$ (4)
```

```
20  PRINT A$;"BLOAD AB, D2"
30  PRINT A$;"BLOAD BTABLE, D1"
40  CALL 8192
50  END
```

方法 2

① 在浮点BASIC状态下PR≠6或冷启动造型软件包盘

② 进入打印功能模块屏幕出现

请输入名字 AB ↙

进入打印坐标模块打印出:

SHAPE	X0	Y0
1	88	90
2	106	48
3	82	61
4	130	78

③ 在浮点BASIC状态下编程为

```
10  A$=CHR$(4)
20  PRINT A$;"BLOAD AB, D2"
30  FOR I = 1 TO 4: READ X(I), Y(I):
NEXT
40  HGR 2
50  POKE 232, Ø: POKE 233, 96
60  HCOLOR=3: ROT=0: SCALE=1
70  FOR I=1 TO 4
80  DRAW I AT X(I), Y(I): NEXT
90  END
```

100 DATA 88, 91, 106,48, 82, 61, 130,78

例4,已知在图形编辑状态下建立的几幅图形，现需脱离造型软件包顺序显示及拷贝它们。

步骤：

① 调入该图表务必加上后缀. TAB

② BRUN DRAWSG, D1

当按任意键顺序显示，在接通打印机后，按P键可拷贝屏幕显示的图形。

九、注意事项

① 经过实用文件压缩存盘的只是造型表文件首地址不定。在调用该造型表时，务必加上调用到内存的地址。经过压缩存盘的造型表不能再附加新的造型，只能供积木或图形编辑使用，否则会出错。

② 机器若出现死锁，随热启动机器或按恢复键，原来的图表没有丢失，可继续建立造型。勿冷启动（关机），除非您不需内存中的图表。

第 二 篇
实用图形管理与技巧

第一章 初步知识

本章作为第二篇的引导，介绍一些初步知识，既有BASIC语言部分，也有6502机器语言内容。主要有机器语言子程序的存贮和调用，形式多样；图形搬家的方法，应有尽有；高分辨率图形的绘制，技巧性高；图形数据的拟合处理，实用性强。

第一节 机器语言子程序的存贮和调用

在BASIC程序中常常用到一些机器语言子程序，它们是怎样存贮的，又有哪些调用方法？

例如，在显示屏上输出26个英文字母，最简便的方法是编制一个BASIC程序，见程序GP 2101。

```
10 A = 65
20  PRINT  CHR$ (A);
30 A = A + 1
```

```
40   IF A     = 90 THEN 20
50   END

]RUN
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

程序GP2101明显的缺点是占用内存较多。

若改用机器语言编制，只需13个字节，见程序GP2102。

```
*300.30CL

0300-   A9 C1        LDA   #$C1
0302-   20 ED FD     JSR   $FDED
0305-   18           CLC
0306-   69 01        ADC   #$01
0308-   C9 DB        CMP   #$DB
030A-   D0 F6        BNE   $0302
030C-   60           RTS
```

在程序GP2102中，机器语言代码存贮在＄0300开始的一段地址中，首先将A的代码C1放进累加器，调用输出一个字符子程序，其入口地址为＄FDED，然后清进位位，累加器内容加1，判断是不是＄DB（Z的代码是＄DA），不是跳转＄0302，否则返回。其中BNE是相对寻址的转移指令，在Z标志为0时转移，操作码为D0，操作数是相对地址，又称偏移量，其值计算方法是：

＄030A＋＄2＋偏移量＝＄0302

结果偏移量是负值＄0A，需采用补码表示，即F6(256－10＝246，相当于＄F6)。

现在，我们以机器语言子程序GP2102为例，介绍存贮

和调用的方法。

一、用CALL指令

在监控状态下，按入机器语言子程序GP 2102，返回BASIC 后,取名GP2102并存盘：BSAVE GP 2102，A$0300,. L$0D。

```
*300.30C

0300- A9 C1 20 ED FD 18 69 01
0308- C9 DB D0 F6 60
```

① 直接从键盘键入：

] CALL 768↙

② 编制一个BASIC程序并运行之：

```
10 CALL 768
20 END
```

执行上述两种方法，均应先将机器语言子程序调入内存，即BLOAD GP 2102↙。

二、用POKE指令

将机器语言子程序代码转换成十进制数，并放在DATA语句中，然后用CALL指令调用。即编制一个BASIC 程序GP 2103。

```
10  FOR I = 0 TO 12
20  READ X
30  POKE 768 + I,X
40  NEXT I
50  CALL 768
```

```
60   END
70   DATA 169,193,32,237,253,24,105
      ,1,201,219,208,246,96
```

```
]RUN
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

三、用BLOAD命令

直接键入BLOAD GP2102↙，然后用CALL 768↙调用，也可以用程序GP2104调用。

```
10 D$ =  CHR$ (4)
20  PRINT D$;"BLOAD GP2102"
30  CALL 768
40  END
```

四、用&命令

① 源程序（见GP2105）

```
10   PRINT  CHR$ (4);"BLOAD GP2102"

20   POKE 1013,76: POKE 1014,0: POKE
      1015,3
30   &
40   END
```

```
]RUN
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

② 在监控状态下按入

*3F5：4C 00 03↙

返回BASIC后：

] BLOAD GP2102↙

] &↙

③ 源见程序（见GP2106）

```
10   FOR I = 0 TO 12: READ X: POKE
      768 + I,X: NEXT
20   DATA 169,193,32,237,253,24,105
      ,1,201,219,208,246,96
30   POKE 1013,76: POKE 1014,0: POKE
      1015,3
40   CALL 768
50   END
```

④ 键入机器语言，如程序GP2107，并以GP2107存盘备用。在监控状态下运行。

```
0300- 4C 03 03 A9 C1 20 ED FD
0308- 18 69 01 C9 DB D0 F6 60
*300G
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

⑤ 也可以键入机器语言子程序，并以GP2108存盘，既可以在监控状态下运行，也可以在BASIC状态下运行。

```
0300- A9 C1 20 ED FD 18 69 01
0308- C9 DB D0 F6 60 4C 00 03
*300G
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

```
*
[]&
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

在BASIC程序中，当遇到"&"号开头的命令时，会自动跳转到$3F5去执行机器语言子程序。通常在$3F5—$3F7中存放JMP XXXX（一般情况下是4C 58 FF），因此改变$3F6，$3F7中的内容，就可以跳转到改过内容的地址上去。例如，程序GP2105中安排3个POKE指令：POKE 1013，76：POKE 1014，0：POKE 1015，3就相当于：$3F5—4C 00 03，从而转向$0300开始的存贮空间去。

五、设置在REM语句中

对于不太长的机器语言子程序，可以将其放在BASIC程序的REM语句中。这是因为REM语句是非执行语句，BASIC解释程序碰到它会自动跳过，而不访问它的具体内容，因此，REM之后的内容，可以不按语法规则存放。将机器语言子程序嵌入REM语句中，就可以将它和BASIC程序连在一起，两者同时存贮和调用。

为此，必须首先算出机器语言的首地址。

为方便计，可把REM作为BASIC程序中的第一个语句。众所周知，BASIC程序区的首地址是从$0801（即2049）开始的，然后是2个字节的链指针（下行程序首地址，低址在前，高址在后），2个字节的本行行号（低位在前，高位在后）及1个字节的REM语句（本行程序只有1个语句，其内部代码为$132），这样，就可算出机器语言子程序的入口地址：

2049 + 2 + 2 + 1 = 2054　（$806）

因此，BASIC程序中调用机器语言子程序应用CALL 2054。

对于文中开头提到的机器语言子程序GP 2102，可按下述步骤组织：

• 在监控状态下从地址$806开始键入机器语言子程序GP 2102。

• 在BASIC状态下，将上述机器语言子程序取名GP 2109并存入盘中。

• 键入BASIC程序GP 2110：

```
10  REM ABCDE12345678
20  CALL 2054
```

• 调GP 2109的机器语言子程序：

BLOAD GP 2109↙

• 再LIST BASIC程序GP 2110，结果为：

```
10  REM  SPEED=  TO   LG  i  -  RND
       =   '
20  CALL 2054
```

整个过程清单：

```
0806-  A9  C1
0808-  20  ED  FD  18  69  01  C9  DB
0810-  D0  F6  60
```

```
*                    /
]BSAVE GP2109,A$806,L$D

]LOAD GP2110

]LIST

10   REM ABCDE12345678
20   CALL 2054



]BLOAD GP2109


]LIST

10   REM   SPEED=   TO    LG   i  -   RND
        =    '
20   CALL 2054

]RUN
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

可见机器语言子程序已装入第10行的REM语句中。

·最后再把该程序存入盘中，供以后使用。

采用上述方法组织程序时，存放在REM语句中的假想数据个数，应大于或等于机器语言子程序的长度（本例是13个字节，可以放13个字符，也可以按入13个空格），同时，在机器语言子程序编制中应防止$00出现，否则BASIC解释系统会误认为是程序结束，而作出错误的处理。

六、用字符串方式存贮

可以编制一个BASIC程序（见GP 2111）。

```
100 K$ = "300:A9 C1 20 ED FD 18 69
      01 C9 DB DO F6 60 N D823G"
110 L = LEN (K$)
120 FOR I = 1 TO L: POKE 511 + I,
      ASC ( MID$ (K$,I,1)) + 128
130 NEXT I
140 CALL - 144: CALL 768
150 END

]RUN
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

程序GP 2111的优点是不必将机器码转换成十进制数，而是把机器语言子程序作为 BASIC 字符串的形式赋给K $中，再用循环的方法，将机器语言子程序的每一个字符，转换成负 ASCII码，并送入从$0200开始的第二页监控下的键盘缓冲区，再调用键盘扫描子程序（CALL —144）来回收送入其中的信息，从而建立机器语言子程序，因此时仍处在监控状态下，为了返回 BASIC 状态，在字符串K$的最后加了一条D 823监控命令，最后只要在程序中安排CALL 768调用即可。

七、修改某些指针

在 BASIC 程序的结尾处（3个全0字节）以后单元，存放机器语言子程序，而把程序中变量首指针（在$69，$6A单元中查）和程序尾指针（从$AF，$BO单元中查），设置在机器语言子程序之后，这样，用SAVE命令存入磁盘时，能把 BSAIC 程序和机器语言子程序一并存入，而用 LOAD 命令时，又把它们同时装入内存。显然，这样组

织的程序可以减少访问磁盘的次数。

例如，对文中开头提到的例子，可以用下述方法组织：

· 健入BASIC程序：

10 CALL 768

20 END

在监控状态下，可以看到它们的存贮格式：

```
10   CALL 768
20   END

]CALL-151

*801.811

0801- 0A 08 0A 00 8C 37 36
0808- 38 00 10 08 14 00 80 00
0810- 00 00
```

从$0801开始，各地址存贮内容说明如下：

$0801—0A；下行程序首地址低位

$0802—08；下行程序首地址高位

$0803—0A；本行行号低位，行号10

$0804—00；本行行号高位

$0805—8C；本行程序第一语句CALL，内码是8C

$0806—37；字符7的内码是37

$0807—36；字符6的内码是36

$0808—38；字符8的内码是38

$0809—00；00为本行结尾标志

＄080A—10；下行程序首地址低位

＄080B—08；下行程序首地址高位

＄080C—14；本行行号低位，行号20

＄080D—00；本行行号高位

＄080E—80；本行语句END的内码

＄080F—00；本行结束

· 至于机器语言子程序，我们可以放在＄0818开始的单元存放，即在监控状态下，从＄0818开始，连续输入机器语言子程序：

```
0818- A9 C1 20 ED FD 18 69 01
0820- C9 DB D0 F6 60
```

· 显然，原来BASIC程序中调用机器语言的入口地址应该改变。因为原来存贮在＄0300开始单元的机器语言子程序，应用CALL 768调用，而现在机器语言子程序，已改存在＄0818单元,相当于十进制数2072，所以应改为CALL 2072。即：

```
10 CALL 2072
20 END
```

在监控状态下，可以看到对应上述BASIC程序的内容：

```
0801- 0B 08 0A 00 8C 32 30
0808- 37 32 00 11 08 14 00 8C
0810- 00 00 00 00 00 00 00 00
0818- A9 C1 20 ED FD 18 69 01
0820- C9 DB D0 F6 60
```

从中可以看出：＄0801的内容由＄0A改为＄0B，这是因为下行程序的首地址改变了(由于CALL 2072比CALL 768多了一个字节的存贮)，其它还有几种改动，读者可以从对比中看到存贮形式的改变。

·最后应将机器语言的结束地址（本例是＄0824),放入＄AF、＄B0和＄69、＄6A中，即将＄AF、＄B0和＄69、＄6A改为指向＄0824,方法是在监控状态下按入：

AF：24　08

69：　24　08

至此，全部修改指针组织程序的工作完成。返回BASIC状态后，将程序存入盘中，以后只要用LOAD“文件名”调用即可。

综上所述，上述7种方法，各具特点。使用时，可根据需要，灵活选用。用CALL指令的方法最简单，但必须了解机器语言的存贮地址，才能正确调用。如果不知道存贮地址，可以先将机器语言子程序（盘片中有B记号的）调进内存，再从＄AA72（43634)和＄AA73（43635)中查到，即用了PRINT PEEK(43634)+PEEK POKE(43635)*256↙，然后用CALL〈地址〉调用。用POKE命令，其缺点是要将机器语言子程序代码转换成十进制数，优点是直接运行程序即得结果。用BLOAD命令的方法要用DOS命令，这是常用的方法，但对于图形信息，仅用CALL命令还不行，必须加显示图形的程序段。用&命令的方法，比较巧妙，机器语言调进内存后，按&号并回车即可。用设置在REM语句中的方法，将机器语言子程序嵌入REM语句中，和BASIC程序同时存贮、同时调用。用字符串方式

存贮的方法，其特点是不需要将机器语言子程序转换成十进制数，这在图形资料信息的处理中十分有用，即使对图形十分复杂，占用内存很多（如8K）的信息，显示速度也极快。用修改某些指针的方法，采用修改某些指针，从而更好地组织程序，减少对磁盘的磨损，节省时间，这样处理，对程序较为固定又经常使用的，比较好，一些游戏程序，常用此法。

第二节　图形搬家的方法

搬家又称移动，它是指下面3种情况：

·数据资料或图形资料从内存中的一段区域搬至另一段存贮区域。

·机器语言子程序从内存的某个空间搬至另外一个空间。

·内存和磁盘之间信息的相互移动和搬迁到新的地址。

信息（泛指数据资料、图形资料等）的搬移可以用BASIC程序控制，也可以用机器语言子程序控制；可以在BASIC状态下进行，也可以在监控状态下实现。

一、内存和磁盘之间图形资料的搬家

例如，在内存中已有一个图形资料，它存放在高分辨图形第一页内，今将其转移到磁盘中，可执行：BSAVE 文件名，A $6000,L $2000,此时盘片中的图形资料存在地址$6000到$7FFF中。若将上述图形资料搬至内存中高分辨第二页内，用DOS命令比较方便，即：

```
10  D$=CHR$（4）
20  PRINT D$“BLOAD”文件名，A$4000
```

二、内存中图形资料的搬家

例如，将高分辨图形第一页信息搬移到第二页，最简便的方法是执行监控下的“搬家”命令M（MOVE），利用它来移动一批图形资料相当方便，且速度很快，即

]CALL—151 ↙

*4000<2000.3FFFM↙

三、用BASIC程序控制

例如，磁盘中存放在A $2000—A $3FFF中的图形，执行BASIC程序GP2112，即可自动进行调盘、搬家、显示、清屏的全部操作。此时内存中既有原有盘中第一页的图形，又有内存中第二页的图形。

```
10 D$ =  CHR$ (4)
20  INPUT "NETER PICTURE FILE:A1$"
     ;A1$
30  PRINT D$;"BLOAD",A1$
40  GOSUB 90
50  CALL  - 144
60  POKE  - 16299,0: POKE  - 16304
     ,0: POKE  - 16297,0: POKE  -
     16302,0
70  FOR I = 1 TO 2000: NEXT I
80  TEXT : HOME : END
90 K$ = "4000<2000.3FFFM N D823G"
100  FOR I = 1 TO  LEN (K$): POKE
      511 + I, ASC ( MID$ (K$,I,1))
       + 128: NEXT I
110  POKE 72,0: RETURN
```

四、调用机器语言搬家子程序

在中华学习机CEC-I的ROM中（同APPLE机一

样），有一段机器语言子程序（见程序GP 2113），它的功能是专管“搬家”的，其起始地址在$FE2C中，末地址是$FE35，该程序还调用另一个机器语言子程序（见程序GP 2114），被调用的机器语言子程序GP 2114，存贮在$FCB4－$FCC8中。

引导DOS后，在监控状态下，键入上述两个地址范围，会看到两个机器语言各自的存贮内容。

① 源程序（见GP 2113）

```
FE2C-    B1 3C       LDA   ($3C),Y
FE2E-    91 42       STA   ($42),Y
FE30-    20 B4 FC    JSR   $FCB4
FE33-    90 F7       BCC   $FE2C
FE35-    60          RTS
```

② 源程序（见GP 2114）

```
FCB4-    E6 42       INC   $42
FCB6-    D0 02       BNE   $FCBA
FCB8-    E6 43       INC   $43
FCBA-    A5 3C       LDA   $3C
FCBC-    C5 3E       CMP   $3E
FCBE-    A5 3D       LDA   $3D
FCC0-    E5 3F       SBC   $3F
FCC2-    E6 3C       INC   $3C
FCC4-    D0 02       BNE   $FCC8
FCC6-    E6 3D       INC   $3D
FCC8-    60          RTS
```

上述两段机器语言子程序GP 2113和GP 2114，实际上是执行地址3＜地址1.地址2M的移动命令。为此，必须把

相应地址存贮在有关单元中，即：

$42：存放地址3的低位

$43：存放地址3的高位

$3E：存放地址2的低位

$3F：存放地址2的高位

$3C：存放地址1的低位

$3D：存放地址1的高位

然后执行FE2CG↙，在返回BASIC状态后键入CLAA-468↙。

例如，欲将高分辨第一页的信息移至第二页，则在监控下：

*42：00 40↙

*3C：00 20 FF 3F↙

*FE2CG↙

应该指出的是，在执行上述操作前，应在BASIC状态下，将图形资料调入高分辨率第一页。

为了省去在监控下操作的步骤，亦可用BASIC程序来处理，不仅速度快，而且十分方便。例如，将调进内存高分辨第二页的图形，搬至第一页并在第一页显示，可执行程序GP2115。

③ 源程序（见GP2115）

```
10  POKE 60,0: POKE 61,64: POKE 62
     ,255: POKE 63,95
20  POKE 66,0: POKE 67,32
30  CALL  - 468
```

```
40   POKE  - 16304,0: POKE  - 16297
      ,0: POKE  - 16302,0: POKE  -
      16300,0
50   FOR I = 2000: NEXT I: TEXT
60   HOME : END
```

当然在执行本程序GP2115以前，必须调入有关图形，例如BLOAD文件名，A$4000↙。

五、机器语言子程序的搬家

在实际工作中，有时候需要将有用的机器语言子程序，从内存的一段地址搬移到内存的另一段地址去；或者将几段机器语言子程序，合并成一个机器语言子程序，这都属于机器语言子程序的搬家。但是，这种搬家必须注意改动某些关键地址，否则不仅搬家失误，而且面貌全非，程序无法执行。

下面我们通过对上述ROM中的机器语言搬家子程序的搬家，说明机器语言子程序搬家的方法及注意问题。

例如，要求将程序GP2113和程序GP2114合并成一个程序，并搬至$6000开始的一段区域中，然后以MOVE-1存盘。

全部操作按下述步骤执行：

```
*6000<FE2C.FE35M

*600A<FCB4.FCC8M

*6005:0A 60
```

```
*BSAVE
]BSAVE MOVE-1,A$6000,L$20
]CALL 24576
```

六、BASIC语言的搬家程序

程序GP 2116是一个用BASIC语言写的搬家程序，它可以将高分辨第一页的图形数据，搬至第二页去。

① 源程序（见GP 2116）

```
10   FOR I = 768 TO 811
20   READ X
30   POKE I,X
40   NEXT I
50   DATA 169,0,133,96,133,98,169,6
      4,133,99,169,32,133,97,160,0,
      177,96,145,98,200,192
60   DATA 255,208,247,165,99,24,105
      ,1,133,99,165,97,24,105,1,133
      ,97,201,64,144,227,96
70   END
```

为了检验程序GP 2116的搬家本领，可加以下程序段：

```
2 D$ =  CHR$ (4)
4   INPUT "ENTER PICTURE NAME:";K$
6   PRINT D$;"BLOAD"K$",A$2000"
62   HGR2 : POKE  - 16302,0
63   HCOLOR= 7
64   FOR I = 1 TO 500: NEXT I
65   TEXT : HOME
```

这样，调进内存的图形（在高分辨第一页面），经过10—

60的搬家，图形转移到第二页面去，执行62句显示图形，再经64句延迟一段时间，恢复文本、清屏，最后结束。

值得提出的是，运行程序GP 2116，可以在监控下看到一个对应上述BASIC搬家程序的机器语言子程序GP 2117，而且十分有趣的是，这段机器语言子程序同样具有上述搬家的功能。

```
0300-  A9  00  85  60  85  62  A9  40
0308-  85  63  A9  20  85  61  A0  00
0310-  B1  60  91  62  C8  C0  FF  D0
0318-  F7  A5  63  18  69  01  85  63
0320-  A5  61  18  69  01  85  61  C9
0328-  40  90  E3  60
```

② 检验GP 2117机器语言搬家功能的方法

•] RUN GP 2116 ↙

•] NEW↙

•] BLOAD图形文件名，A $ 2000 ↙

•] CALL 768 ↙

•] POKE -16299，0：POKE -16304，0：
  POKE -16297，0：POKE -16302，0

第三节　高分辨率图形的绘制

下面几个程序，都是由高分辨率作图语句绘制的图象。

一、图象绘制和打印

程序GP 2118是一个少女的图象（见图2.1），运行程序

后，首先输入尺寸SIZE，如取1，2，或3，再送入颜色COLOR代码，在1—7之间，则屏幕上显示出不同颜色，不同大小的人像，当人像完全显示后，回到文本状态，如要打印，则按“Y”并回车，反之按任意键结束。



图 2.1

程序GP2118清单。

```
5  PRINT : PRINT : PRINT : PRINT :
      PRINT
10  INPUT " SIZE=";U
20  INPUT " COLOR=";C
100 N = O: PRINT : PRINT : PRINT :
      PRINT
200  HGR2
205  HCOLOR= C
220  READ X
230  IF X < 0 THEN N = N + 1: GOTO
     220
240  IF X > 36 THEN  FOR XX = 1 TO
     2000: NEXT : TEXT : HOME : GOTO
     10000
```

```
250  READ Y
265  HPLOT U * X + 50,N TO Y * U +
     50,N:N = N + 1
270  MUSIC N,10
290  GOTO 220
300  DATA  14.2,18,-1,11.8,19.7,-1
     ,10.5,20,-1,10
301  DATA   22.0,-1,9.4,22.8,-1,9,
     23,-1,8.6,23,-1,9.2,22.9,-1,9
     .9,22.7
302  DATA  -1,10,22.5,-1,10,22.3,-
     1,9.7,22.2,-1,9.6,21.8,-1,10.
     6,20.8,-1,11,20.6,-1
303  DATA   11.4,20.7,-1,11.5,20.7
     ,-1,15,20.8,-1,14.9,21.3,-1,1
     4.9,21.8,-1
304  DATA  14.9,22.8,-1,14.7,22.9,
     -1,14.5,24,-1,14.2,24.5,-1,13
     .4,24.8,-1,13.1,25.8
305  DATA  -1,12.3,25.9,-1,11.7,25
     .7,-1,11.3,26,-1,10,26.2,-1,9
     .7,26.7
306  DATA  -1,9.7,27.0,-1,10,27.5,
     -1,10.3,27.8,-1,10.7,28.2,-1,
     11.0,28.7,-1,11.4,29.1
307  DATA  -1,11.8,23.2,24.4,29.4,
     -1,12,22.7,24.7,29.6,-1,12.6,
     22.7,25,29.8,-1
308  DATA  12.8,22,24.6,29.4,-1,12
     .8,21.4,24.1,29,-1,12.9,21.2,
     23.8,28.4,-1
309  DATA  13,20.9,23.4,28,-1,12.9
     ,20.7,23,27.4,-1,12.8,20.7,22
     .7,26.8,-1
310  DATA  12.7,21,22,26.3,-1,12.5
     ,25.7,-1,12.2,25.3,-1,11.8,24
     .4,-1,11.5,23.2
```

```
320  DATA  -1,11.2,23.1,-1,10.5,23
     .5,-1,9.7,23.7,-1,9.4,23.9,-1
     ,9,24.3,-1,8.6,24.7
330  DATA  -1,8.2,25.1,-1,7.8,25.5
     ,-1,7.4,25.9,-1,7,26.3,-1,6.6
     ,26.7,-1,6.2,27.1,-1
3440  DATA  5.8,27.5,-1,5.5,27.8,-
     1,5.2,28.1,-1,4.9,28.4,-1,4.6
     ,28.7,-1,4.3,29.0,-1
3550  DATA  4.1,29.2,-1,3.9,29.4,-
     1,3.7,29.6,-1,3.5,29.8,-1,3.4
     ,29.9,-1,3.3,30.0,-1
3660  DATA  3.2,30.1,-1,40
10000  INPUT " DO YOU PRINT ? ";A$

10010  IF A$ = "Y" THEN  PR# 1: POKE
      1913,2: PRINT  CHR$ (17)
10020  PR# 0
10030  END
```

二、折线图形绘制

程序GP 2119可以完成折线图形的绘制，20句绘Y轴，40句绘X轴，50—95句点X轴上的坐标，100—200句点Y轴上的坐标。全部数据（X，Y值）放在390—400句的DATA语句中，用290—360句的循环完成各点的绘制工作。

程序GP 2119清单。

```
10  HGR2 : HCOLOR= 7
20  HPLOT 5,180 TO 279,180
40  HPLOT 10,185 TO 10,0
50  FOR X = 30 TO 270 STEP 20
```

```
60  HPLOT X,180 TO X,175
95  NEXT X
100  FOR Y = 160 TO 20 STEP  - 20
110  HPLOT 10,Y TO 15,Y
200  NEXT Y
290 N = 50
310  DIM X(50),Y(50)
315 X(0) = 16:Y(0) = 180 - 20.44
320  FOR K = 1 TO N
330  READ X(K),Y(K)
335  IF X(K) =  - 1 AND Y(K) =  -
     1 THEN  FOR I = 1 TO 3000: NEXT
     : TEXT : END
340 X(K) = (X(K) * 10 + 10) * 8 -
     70
350 Y(K) = 180 - Y(K) * 4
355  HPLOT X(K),Y(K) TO X(K - 1),Y
     (K - 1)
360  NEXT K
390  DATA  0.05,5.11,0.20,30.72
400  DATA  0.57,5.20,1.017,23.74,1
     .47,6.41,1.55,0.39,1.6,7.68,1
     .67,34.18,1.78,22.99,1.88,9.7
     4,2.03,11.1,2.15,11.6,2.28,5.
     96,2.32,1.09,2.42,14.1,2.57,3
     5.9,2.72,33.8,2.95,15.31,3.08
     ,2.86,3.18,3.77,3.28,3.83
500  DATA -1,-1
```

运行结果见图2.2。

三、汉字图形绘制

程序GP 2120是一个BASIC程序，它用高分辨作图语句绘出“南京大学”4个大字。该程序设计思想新颖，难度较大，但从中可以学会用高分辨作图语句绘制汉字的方法。

完整的程序清单见GP 2120，下面是运行结果（见图2.3)。

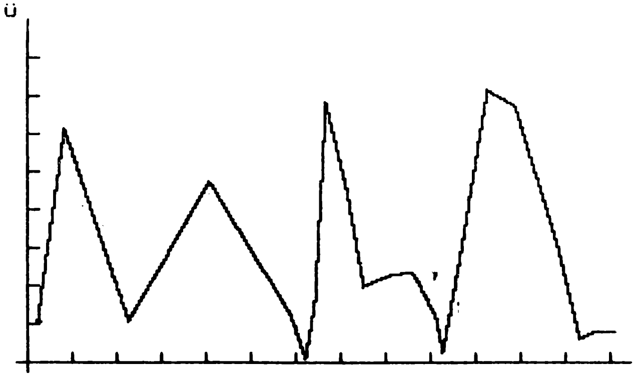


图 2.2

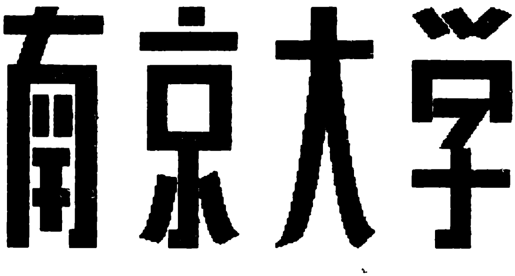


图 2.3

程序GP 2120清单。

```
10   HGR2
20   HCOLOR= 3
```

```
30  FOR I = 10 TO 60
40  HPLOT I,30 TO I,40
50  HPLOT 10 + 0.2 * I,60 TO 10 +
     0.2 * I,170
60  HPLOT 60 - 0.2 * I,60 TO 60 -
     0.2 * I,170
70  HPLOT 0.7 * I + 5,60 TO 0.7 *
     I + 5,70
80  HPLOT 22.5 + 0.35 * I,110 TO 2
     2.5 + 0.35 * I,120
90  HPLOT 22.5 + 0.35 * I,135 TO 2
     2.5 + 0.35 * I,145
95  HPLOT 43 + 0.1 * I,160 TO 43 +
     0.1 * I,170
100 X = 15 + 0.2 * I
110  HPLOT X,67 - 0.7 * I TO X + 1
     0,67 - 0.7 * I
120  HPLOT 27.5 + 0.2 * I,120 TO 2
     7.5 + 0.2 * I,160
130  HPLOT 23.7 + 0.15 * I,78 TO 2
     3.7 + 0.15 * I,102
140  HPLOT 35 + 0.15 * I,78 TO 35 +
     0.15 * I,102
150  NEXT I
200  FOR I = 0 TO 50 STEP 0.1
210  HPLOT I + 80,40 TO I + 80,50
220  HPLOT 100 + 0.25 * I,25 TO 10
     0 + 0.25 * I,35
230  HPLOT 84 + 0.2 * I,60 TO 84 +
     0.2 * I,110
240  HPLOT 126 - 0.2 * I,60 TO 126
      - 0.2 * I,110
250  HPLOT 85 + 0.8 * I,100 TO 85 +
     0.8 * I,110
260  HPLOT 85 + 0.8 * I,60 TO 85 +
     0.8 * I,70
```

```
270  HPLOT 100 + 0.2 * I,110 TO 10
     0 + 0.2 * I,170
275  HPLOT 100 + 0.2 * I,110 TO 10
     0 + 0.2 * I,170
280  HPLOT 95 + 0.2 * I,160 TO 95 +
     0.2 * I,170
285 X1 = (80 + 0.2 * I):Y1 = 0.015
      * I * I
290  HPLOT X1,164 - Y1 TO X1 + 9,1
     67 - Y1
295  HPLOT 130 - 0.21 * I,162 - Y1
      TO 121 - 0.21 * I,168 - Y1
300  HPLOT I + 150,45 TO I + 150,5
     5
310 Y1 = 0.05 * I * I
320  HPLOT 160 + 0.4 * I,166 - Y1 TO
     150 + 0.4 * I,170 - Y1
340  HPLOT 200 - 0.4 * I,160 - Y1 TO
     192 - 0.4 * I,170 - Y1
350  HPLOT 168 + 0.26 * I,25 TO 16
     8 + 0.26 * I,45
360  NEXT I
500  FOR I = 0 TO 50 STEP 0.1
503 X1 = 220 + 0.2 * I:Y1 = 25 + 0
     .3 * I
505 X2 = 270 - 0.26 * I:Y2 = 25 +
     0.30 * I
510  HPLOT X1,Y1 + 4 TO X1 + 9,Y1
520  HPLOT X2,Y2 + 4 TO X2 - 9,Y2
530  HPLOT X1 + 15,Y1 + 4 TO X1 +
     24,Y1
550  HPLOT 220 + 0.2 * I,57 TO 220
      + 0.2 * I,93
560  HPLOT 220 + I,57 TO 220 + I,6
     7
```

```
565  HPLOT 270 - 0.2 * I,57 TO 270
      - 0.2 * I,93
570  HPLOT 231 + 0.4 * I,80 TO 231
      + 0.4 * I,90
580 X3 = 252 - 0.35 * I:Y3 = 80 +
     0.6 * I
590  HPLOT X3,Y3 TO X3 + 10,Y3
600  HPLOT 240 + 0.2 * I,110 TO 24
     0 + 0.2 * I,170
610  HPLOT 220 + I,123 TO 220 + I,
     133
620  HPLOT 232 + 0.25 * I,160 TO 2
     32 + 0.25 * I,170
650  NEXT I
```

值得提出的是程序GP2120，用作图语句写汉字，是一个新的尝试，程序的编制水平较高，应仔细阅读，才能理解。

第四节　图形数据的拟合过程

通过计算机绘制一般的工程图样及一些比较有规律的曲线，相对于复杂曲面轮廓和一些不规则的曲线来说，要困难一些。但是，可以通过数据拟合较好地解决。所谓数据拟合，就是利用曲线对给定的一组数据进行插值和逼近的问题。

我们这里省去复杂的原理和严格的数学证明，而是提供一个有价值的拟合曲线实用子程序。这样，用户只要输入不同的图形数据，而由计算机自动处理复杂曲线的绘制问题。

图 2.4 是用程序GP 2121绘制的，对于一个完整的连续线段构成的图形，程序GP 2121是一个常用的方法。它把一个飞机轮廓曲线基本上勾画出来了，飞机数据资料全部放在

DATA语句中。但是我们看到图形不美观，线条不光滑，加上包括APPLE Ⅱ在内的所有兼容机（也含中华学习机），即使在高分辨率作图下，由于分辨率还不是太高，图形的折线突出的现象几乎是不可避免的。



图 2.4

程序GP 2121清单。

```
10  HGR2 : HCOLOR= 7
310  DIM X(50),Y(50)
315 X(0) = 237:Y(0) = 165
320  FOR K = 1 TO 50
330  READ X(K), (K)
340  IF X(K) =  - 1 AND Y(K) =  -
     1 THEN  FOR 1 = 1 TO 5000: NEXT
     I: TEXT : HOME : END
355  HPLOT X(K),191 - Y(K) TO X(K -
     1),191 - Y(K - 1)
360  NEXT K
390  DATA 221,165,218,165,204,147,
     191,130,162,131,52,138,33,128
     ,0,111,48,103,70,100,88
400  DATA 102,102,104,111,111,120,
     118,200,118,218,118,230,127,2
     20,136,237,165
500  DATA -1,-1
```

图2.5是程序GP 2122绘制出来的，和图2.4相比，图形

线条明快，光滑美观。主要进行了对图形数据的拟合处理。

程序GP 2122清单。

```
10  HGR2
20  HCOLOR= 7
30 E1 = 8:E2 = 0
40  READ X2,Y2
50 X0 = X2:Y0 = Y2
60  READ X1
70  IF X1 < 500 THEN  READ Y1,X2,Y
    2: GOSUB 1000: GOTO 50
80 X = 221:Y = 165
90  HPLOT  TO X,191 - Y
100  END
110  DATA 221,165,218,165,204,147,
     191,130,162,131
120  DATA  52,138,33,128,0,111,48,
     103,70,100,88
130  DATA 102,102,104,111,111
140  DATA 120,118,200,118,218
150  DATA 118,230,127,220,136,237,
     165,500
1000 X = X0:Y = Y0
1010  HPLOT X,191 - Y
1020 A1 =  - 2 * X0 + (2 + E1) * X
     1
1030 A2 = X0 - (2 + E1) * X1 + (1 +
     E1 + E2) * X2
1040 B1 =  - 2 * Y0 + (2 + E1) * Y
     1
1050 B2 = Y0 - (2 + E1) * Y1 + (1 +
     E1 + E2) * Y2
1060  FOR I = 0 TO 1 STEP .025
1070 G = 1 + E1 * I + E2 * I * I
```

```
1080 X = (X0 + A1 * I + A2 * I * I
     ) / G
1090 Y = (Y0 + B1 * I + B2 * I * I
     ) / G
1100  HPLOT  TO X,191 - Y
1110  NEXT
1120  RETURN
```



图 2.5

值得指出的是，若用绘图仪绘出经拟合后的曲线，画面将更加美观，线条非常光滑，没有任何一点不连续感。

第二章 图形变换

图形变换包括图形移动、放大、缩小、旋转、剪辑等方面内容，它们被广泛运用到计算机图形学的各种应用程序中。本章主要介绍图形移动、放大、缩小、镜像复制和简单的图形动画原理和方法，并提供一些示范程序。关于图形剪辑部份的内容，我们放在本篇第九章中介绍。

第一节 图形的移动

本节通过图形移动的实例，介绍图形变换的基本原理、方法及其实现步骤。为便于理解，采用BASIC语言编写程序。

图形移动的实质是把组成该图形的各顶点坐标，作如下变换：

$$XT = X + H$$
$$YT = Y + V$$

式中X、Y为移动前图形上某一点的坐标；XT、YT为移动后该点的坐标；H、V则分别为水平和垂直方向要移动的距离。

当经过上式坐标变换后，原来的图形将被移到一个新的位置上，输入不同的参数H和V，则屏幕上将出现不同位置的图形（不清屏时）。如果原来显示的图形全部清除，则保留在新的位置上的图形。这样，就实现了图形的移动。

一、矩形图形的移动

程序GP 2201可以实现一个矩形图形的移动(见图2.6)，不论是原来图形还是移动后的图形，都显示在同一个画面上(即不清屏，或者说不抹除原有图象)。

```
50   ONERR  GOTO 300
60   HGR : HCOLOR= 7
70   DIM X(1,30),Y(1,30),N(1)
80 XM = 279:YM = 191
110 PN = 1:E = 0
120  FOR I = 1 TO 100
130  READ XD,YD
140 E = E + 1
150 X(PN,E) = XD
160 Y(PN,E) = YD
170  IF XD =  - 1 AND YD =  - 1 THE
     185
180  NEXT I
185 N(1) = E
190  PRINT : PRINT
200  FOR K = 1 TO PN
210  FOR J = 1 TO N(K) - 2
220  HPLOT X(K,J),Y(K,J) TO X(K,J +
     1),Y(K,J + 1)
230  NEXT J: NEXT K
240  POKE  - 16301,0
300  INPUT H,V
302  IF H =  - 99 AND V =  - 99 THEI
      END
305  IF H >  XM OR V > YM THEN 300
310  FOR J = 1 TO N(1)
320 X(1,J) = X(1,J) + H
330 Y(1,J) = Y(1,J) + V
340  NEXT J
```

```
400  FOR K = 1 TO PN
410  FOR J = 1 TO N(K) - 2
420  HPLOT X(K,J),Y(K,J) TO X(K,J +
     1),Y(K,J + 1)
430  NEXT J: NEXT K
440  GOTO 300
500  DATA 20,120,70,120,70,150,20,
     150,20,120,-1,-1
```

这里给出一个运行实例。

```
?20,-40
?20,-40
?20,40
?20,40
?-99,-99


UPOKE1913,1

Ü
```

程序中50句和305句，为防止移动的图形超出屏幕范围而设置的，当用户输入两个－99时，程序运行结束。60—180句，是将原图所有坐标依次读入两维数组X（PN，E），Y（PN，E）中。185—230句为显示原图象。300—340句进行坐标变换，图象转移。400—430句显示移动到新位置的图象。300句为输入水平及垂直方向要求移动的距离。440句跳转到300句，可以连续输入新的参数，从而产生不同位置的多幅图象。

二、简单的动画显示

若将程序GP 2201稍作改动，便产生动画的效果，见程序GP 2202，其中302句HGR命令是清除屏幕，抹除原来显示的图形，然后不断循环，再显示新坐标图形，再重复抹除旧图形，显示新图形，结果看到画面的动态变化。

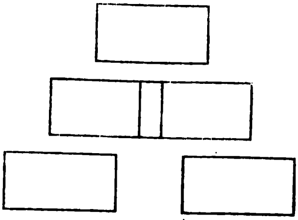


图 2.6

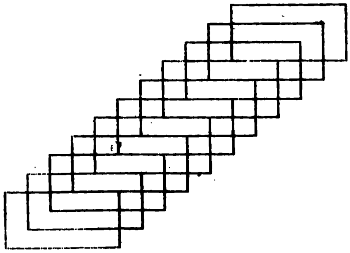


图 2.7

若删去302句，HGR，则看到的是一个连续叠加的图形，见图2.7。

修改后的程序，见GP 2202。

```
50  ONERR  GOTO 300
60  HGR : HCOLOR= 7
70  DIM X(1,30),Y(1,30),N(1)
80 XM = 279:YM = 191
110 PN = 1:E = 0
120  FOR I = 1 TO 100
130  READ XD,YD
140 E = E + 1
150 X(PN,E) = XD
160 Y(PN,E) = YD
170  IF XD = - 1 AND YD = - 1 THEN
     185
```

```
180  NEXT I
185 N(1) = E
190  PRINT : PRINT
200  FOR K = 1 TO PN
210  FOR J = 1 TO N(K) - 2
220  HPLOT X(K,J),Y(K,J) TO X(K,J +
     1),Y(K,J + 1)
230  NEXT J: NEXT K
300  FOR I = 1 TO 10
301  READ H,V
302  HGR
305  IF H > XM OR V > YM THEN 300
310  FOR J = 1 TO N(1)
320 X(1,J) = X(1,J) + H
330 Y(1,J) = Y(1,J) + V
340  NEXT J
400  FOR K = 1 TO PN
410  FOR J = 1 TO N(K) - 2
420  HPLOT X(K,J),Y(K,J) TO X(K,J +
     1),Y(K,J + 1)
430  NEXT J: NEXT K
440  NEXT I
500  DATA 20,120,70,120,70,150,20,
     150,20,120,-1,-1
600  DATA  10,-10,10,-10,10,-10,10
     ,-10,10,-10,10,-10,10,-10,10,
     -10,10,-10,10,-10
```

三、多幅图形的移动

如果希望在屏幕上同时显示3至5个图象（或更多），并用交互式会话方式，由用户选择其中任意一个图象，在水平或垂直方向移动，则画面更加丰富多采，今给出3个图象移动的程序，读者不难加以扩展，并运用其它图形变换技巧，使图形显示更加饶有趣味。

程序中给出3个图形示例：第一个图形是一个矩形长条，其图形轮廓数据放在600句的DATA语句中；第二个图形为一个三角形，相应数据放在700句的DATA语句；第三个图形为一个小动物，其图形数据放在810句和820句的DATA语句中。3组数据之间采用两个－1分隔，图形数据的终止符采用连续两个－100。这种安排保证了该程序与图形的相对独立性。

程序运行方法比较简单，用RUN↙，屏幕上立即看到以下3幅图形，见图2.8。

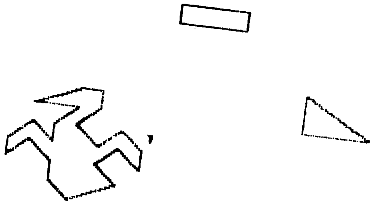


图 2.8

然后若按入3并回车，再按入150，－40↙则看到小动物移动，如图2.9所示。

若按入1并回车，再键入110，70↙则看到矩形长条移到小动物的下部，见图2.10。

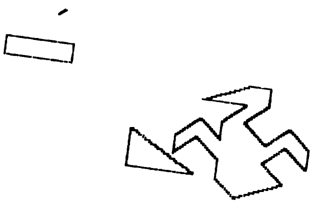


图 2.9

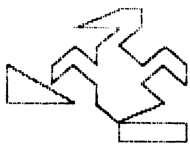


图 2.10

若按入2并回车，再按入－50，－50↙，则三角形离开

小动物远去，见图2.11。

若按入 3 并回车，再按入－20，－30 ↵，则小动物离开矩形长条，向三角形飞去，如图2.12所示。

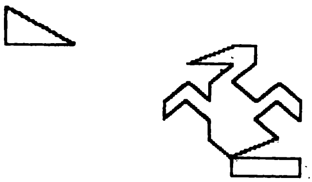


图 2.11

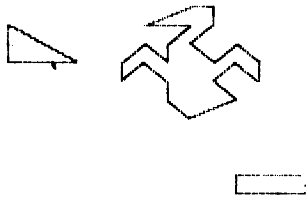


图 2.12

而当按入0，0 时，则程序运行结束。

读者可以模仿上述操作，随心所欲地进行图形的各种移动，如果输入的数据H或V 值不当，以致被移动的图形超出屏幕范围时，则程序自行结束（见460 END）。

本程序图形的个数还可以增加，只要预先在方格纸上画好图形，读取坐标，然后置这些数据于DATA语句中，数据的安排仍和上面规定一样，每两组数据之间，以两个连续－1分隔，不要忘记最后一个图形数据是用两个－100结束。

完整的程序GP 2203清单。

```
60  HGR2 : HCOLOR= 3
70  DIM X(5,100),Y(5,100),N(5)
80 XM = 279:YM = 191
110 PN = 1
120 E = 0
130  READ XD,YD
```

```
140  IF XD     = 0 THEN 200
160 N(PN) = E
170  IF XD =  - 100 THEN 240
180 PN = PN + 1
190  GOTO 120
200 E = E + 1
210 X(PN,E) = XD
220 Y(PN,E) = YD
230  GOTO 130
240  PRINT
250  GOSUB 510
290  PRINT "1-    2-     3-"
310  INPUT P
330  IF P = 0 THEN  TEXT : HOME : END

370  PRINT "H AND V AMOUNT TO MOVE
     "
380  INPUT H,V
410  FOR J = 1 TO N(P)
420 X(P,J) = X(P,J) + H
430 Y(P,J) = Y(P,J) + V
440  IF X(P,J) >  = 0 AND X(P,J) <
      = XM AND Y(P,J) >  = 0 AND Y
     (P,J) <  = YM THEN 470
460  END
470  NEXT J
480  GOSUB 510
490  GOTO 290
510  HGR2
530  FOR K = 1 TO PN
540  FOR J = 1 TO N(K) - 1
550  HPLOT X(K,J),Y(K,J) TO X(K,J +
     1),Y(K,J + 1)
560  NEXT J
570  NEXT K
580  RETURN
```

```
600  DATA  90,40,120,40,120,50,90,
     50,90,40
610  DATA -1,-1
700  DATA 150,80,180,100,150,100,1
     50,80
710  DATA -1,-1
810  DATA 20,130,20,120,30,110,40,
     120,40,110,50,100,30,100,50,9
     0,60,90,60,100,50,110,60,120,
     70,110,80,120,80,130,70,120,6
     0,130,70,140,50,150,40,140,40
     ,130,30,120
820  DATA  20,130
910  DATA  -100,-100
```

第二节 图形的放缩

图形放缩系指图形放大和缩小，它也属于基本的图形变换。和图形移动的设计思想一样，首先必须搞清基本原理，然后再安排程序。

按比例放大和缩小图形，必须首先确定一个参考点的位置。一般情况下，参考点可以设置在图形的左下角，或者安排在图形的中心。设参考点的坐标为 XF、YF，则有以下变换公式：

$$XS = X \cdot HS + XF(1 - HS)$$

$$YS = Y \cdot VS + YF(1 - VS)$$

式中X、Y为变换前图形中某一点的坐标；XS、YS为变换后该点的新坐标；XF、YF为参考点坐标；HS、VS则分别表示为水平方向和垂直方向图形变换（放大、缩小）的比例系数，它们必须为正。显然，当HS>1，VS>1时，表

示图形放大；当HS＜1，VS＜1时，则表示图形缩小。

一、图形的放大和缩小

下面，我们通过在屏幕上显示一辆汽车的实例，观察图形放大或缩小的变化。

汽车图形的轮廓分成3个部分：第一部是FRONT，其图形数据放在920—930句的DATA语句中；第二部分是BODY，相应数据放在950句中；第三部分是REAR，其图形数据放在980—985句中。每一部分数据之间采用连续两个－1分隔，而图形数据最后一部分，则连用两个－1作为结束标志。这样处理，保证了程序与图形的相互独立性。

程序共分4段：

第一段：60—230句为将小汽车全部信息读入两维数组X(PN, E)、Y(PN, E)中。

第二段：770—830句为画图子程序。

第三段：860—890句为坐标变换的子程序。

第四段：360—510句为总控。可以根据输入的P值(1,2,3,4)决定汽车的某一部分或整个汽车进行放大或缩小；而放大和缩小的比例由HS、VS（键盘输入）确定。而对整体部分的放大或缩小，还应输入XF、YF的值（参考点坐标）。结束程序运行输入0,0。

① 源程序（见GP2204）

```
60  HGR2 : HCOLOR= 3
70  DIM X(5,100),Y(5,100),N(5)
80 XM = 279:YM = 191
110 PN = 1
120 E = 0
```

```
130  READ XD,YD
140  IF XD >  = 0 THEN 200
160 N(PN) = E
170  IF XD =  - 100 THEN 240
180 PN = PN + 1
190  GOTO 120
200 E = E + 1
210 X(PN,E) = XD
220 Y(PN,E) = YD * 5 / 6
230  GOTO 130
240  PRINT
290  GOSUB 770
330  PRINT "1-FRONT 2-BODY 3-REAR
     4-ALL 0-END"
360  INPUT P
400  IF P = 0 THEN  TEXT : HOME : END

430  INPUT HS,VS
510  ON P GOTO 560,590,620,690
560 XF = X(1,2)
570 YF = Y(1,2)
580  GOTO 640
590 XF = X(2,2)
600 YF = Y(2,2)
610  GOTO 640
620 XF = X(3,2)
630 YF = Y(3,2)
640  GOSUB 860
650  GOSUB 770
660  GOTO 330
690  INPUT XF,YF
700  FOR P = 1 TO PN
710  GOSUB 860
720  NEXT P
730  GOSUB 770
740  GOTO 330
```

```
770  HGR2
790  FOR K = 1 TO PN
800  FOR J = 1 TO N(K) - 1
805  IF X(K,J) < 0 OR X(K,J) > XM OR
     Y(K,J) < 0 OR Y(K,J) > YM THEN
      TEXT : HOME : END
810  HPLOT X(K,J),Y(K,J) TO X(K,J +
     1),Y(K,J + 1)
820  NEXT J
830  NEXT K
840  RETURN
860  FOR J = 1 TO N(P)
870 X(P,J) = X(P,J) * HS + XF * (1
      - HS)
880 Y(P,J) = Y(P,J) * VS + YF * (1
      - VS)
890  NEXT J
900  RETURN
910  REM  FORNT OF CAR
920  DATA 220,90,210,90,210,60,250
     ,65,260,90,250,90
930  DATA  250,100,240,110,230,110
     ,220,100,220,90,230,80,240,80
     ,250,90
935  DATA -1,-1
940  REM MIDBODY
950  DATA 210,90,160,90,160,60,205
     ,60,187,35,160,35,160,30,190,
     30,210,60
960  DATA -1,-1
970  REM BACK OF CAR
980  DATA 120,90,160,90,160,60,120
     ,60,120,60,135,35,160,35,160,
     30,130,30,115,60
985  DATA 80,60,75,90,90,90,90,100
     ,100,110,110,110,120,100,120,
```

```
      90,110,80,100,80,90,90,90,100
990  DATA -100,-100
```

② 运行实例

]RUN↙后，屏幕上出现一个汽车，该汽车没有经过放大、缩小处理，见图2.13。

?4↙

?0.5，0.5↙

?120，80↙

出现一个缩小一半的汽车（指对原汽车缩小一半），见图2.14。

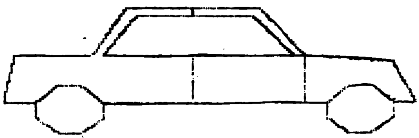


图 2.13

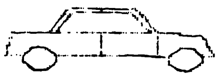


图 2.14

?4↙

?1.2，1.2↙

?100，120↙

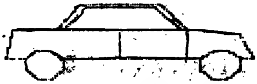


图 2.15

出现一个放大1.2倍的汽车（指对刚才缩小的汽车），见图2.15。

?0，0↙

结束。

将770句HGR2改为PRINT，重新运行程序，并按下述步骤操作，可看到屏幕上4辆汽车的图形，见图2.16。

```
]RUN↙
?4↙
?0.3, 0.3↙
?-25, 140↙
?4↙
?0.8, 0.8↙
?-8, -80↙
?4↙
?0.6, 0.6↙
?-3, -40↙
```

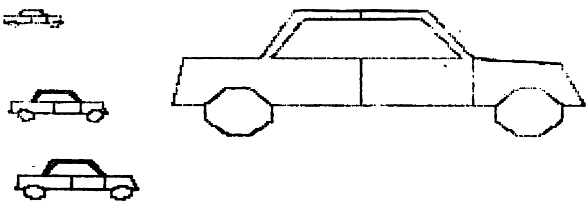


图 2.16

结束运行，输入0，0并回车。

二、图形放缩的动态变化

如果，希望既看到图象的放大或缩小的变化，又看到图象的动态变化，则只要将上述程序GP 2204作一些修改。主要改动原740句为：FOR K=1 TO 2500：NEXT K：GOTO2000，并增加2000—2080的程序段。

修改后的程序取名GP2205。

```
60  HGR2 : HCOLOR= 3
70  DIM X(5,100),Y(5,100),N(5)
80 XM = 279:YM = 191
110 PN = 1
120 E = 0
130  READ XD,YD
140  IF XD >  = 0 THEN 200
160 N(PN) = E
170  IF XD =  - 100 THEN 240
180 PN = PN + 1
190  GOTO 120
200 E = E + 1
210 X(PN,E) = XD
220 Y(PN,E) = YD * 5 / 6
230  GOTO 130
240  PRINT
290  GOSUB 770
330  PRINT "1-FRONT 2-BODY 3-REAR
     4-ALL 0-END"
360  INPUT P
400  IF P = 0 THEN  TEXT : HOME : END

430  INPUT HS,VS
510  ON P GOTO 560,590,620,690
560 XF = X(1,2)
570 YF = Y(1,2)
580  GOTO 640
590 XF = X(2,2)
600 YF = Y(2,2)
610  GOTO 640
620 XF = X(3,2)
630 YF = Y(3,2)
640  GOSUB 860
650  GOSUB 770
660  GOTO 330
690  INPUT XF,YF
700  FOR P = 1 TO PN
```

```
710  GOSUB 860
720  NEXT P
730  GOSUB 770
740  FOR K = 1 TO 2500: NEXT K: GOTO
     2000
770  HGR2
790  FOR K = 1 TO PN
800  FOR J = 1 TO N(K) - 1
805  IF X(K,J) < 0 OR X(K,J) > XM OR
     Y(K,J) < 0 OR Y(K,J) > YM THEN
      TEXT : HOME : END
810  HPLOT X(K,J),Y(K,J) TO X(K,J +
     1),Y(K,J + 1)
820  NEXT J
830  NEXT K
840  RETURN
860  FOR J = 1 TO N(P)
870 X(P,J) = X(P,J) * HS + XF * (1
      - HS)
880 Y(P,J) = Y(P,J) * VS + YF * (1
      - VS)
890  NEXT J
900  RETURN
910  REM  FORNT OF CAR
920  DATA 220,90,210,90,210,60,250
     ,65,260,90,250,90
930  DATA  250,100,240,110,230,110
     ,220,100,220,90,230,80,240,80
     ,250,90
935  DATA -1,-1
940  REM MIDBODY
950  DATA 210,90,160,90,160,60,205
     ,60,187,35,160,35,160,30,190,
     30,210,60
960  DATA -1,-1
970  REM BACK OF CAR
```

```
980  DATA 120,90,160,90,160,60,120
     ,60,120,60,135,35,160,35,160,
     30,130,30,115,60
985  DATA 80,60,75,90,90,90,90,100
     ,100,110,110,110,120,100,120,
     90,110,80,100,80,90,90,90,100

990  DATA -100,-100
1000  DATA 120,80,122,78,124,76,1
     26,74,128,72,128,70,130,68,13
     2,66,134,64,136,64
2000  FOR I = 1 TO 10
2010  READ XF,YF
2020  HGR2
2030  FOR P = 1 TO PN
2040  GOSUB 860
2050  NEXT P
2060  GOSUB 770
2070  NEXT I
2075  FOR K = 1 TO 1000: NEXT K
2080  TEXT : HOME : PRINT "GOOD BY
     E!": END
```

GP2205程序运行后，首先出现一个原数据的汽车图形，然后输入：

?4↙

?0.9, 0.9↙

?120, 80↙

即可在屏幕上看到10个变化的汽车图形，一面一次一次地缩小，一面一次一次地移动，最后自动结束。

至于GP2204和GP2205两个程序局部放大或缩小，输入的P值为1，或2，或3。读者可以自行试验，从中体会程序的

编制原理，以及欣赏图形变换的乐趣。

第三节 镜象复制

在图形显示技术中，使用窗口和视见区的概念，常常可以得到出乎意料的特性。如果我们把视见区的参数在一定范围内倒置，可以得到复制的对称图象，就象我们在镜子里看到一个物体时一样。这种能用较少的数据（把它放在DATA语句中），画出具有一个或几个对称轴的图象，在绘图技术中是很有用的。它可以减少大量重复劳动，绘制出各种形态各异又完全对称的图象。

改变V1、V2、V3、V4的参数，可以得到不同区域的视见区矩形框；在一个屏幕上可以设置几组不同参数的Vi值（i=1,2,3,4)，可以得到几个视见区；而把几个视见区巧妙地组合在一起，就会使视见区内的画面对称。

由于视见区可以复现窗口内的内容，因而改变窗口的大小，可使视见区的图象发生放大和缩小的作用。

下面几个程序实例，同样用到在图形剪辑中的几个子程序（见第二篇第九章）。

一、4个对称的图象

① 程序（见GP2206）

```
130  TEXT : HOME
140 W1 = 0:W2 = 50:W3 = 0:W4 = 25
150  HGR2 : HCOLOR= 3
155  READ V1,V2,V3,V4: GOSUB 60000

160  GOSUB 63500: GOSUB 500
```

```
180  READ V1,V2,V3,V4: GOSUB 60000

190  GOSUB 63500: GOSUB 500
200  READ V1,V2,V3,V4: GOSUB 60000

210  GOSUB 63500: GOSUB 500
220  READ V1,V2,V3,V4: GOSUB 60000

230  GOSUB 63500: GOSUB 500
240  PRINT  CHR$ (7): END
300  DATA   0,69.5,0,47.5
310  DATA   139.5,69.5,0,47.5
320  DATA   0,69.5,90.5,47.5
330  DATA   139.5,69.5,90.5,47.5
500 X = 2:Y = 22:W$ = "U"
505  GOSUB 62300:W$ = "D"
510 X = 47:Y = 22: GOSUB 62300
520 X = 47:Y = 2: GOSUB 62300
530 X = 32:Y = 2: GOSUB 62300
540 X = 32:Y = 7: GOSUB 62300
550 X = 17:Y = 7: GOSUB 62300
560 X = 17:Y = 17: GOSUB 62300
570 X = 2:Y = 17: GOSUB 62300
580 X = 2:Y = 22: GOSUB 62300
590  RETURN
```

② 运行结果（见图2.17）

程序GP2206运行后，在屏幕的左下方得到4个对称的图象，此时视见区较少，4个视见区只占屏幕的四分之一。

二、窗口加大，图形变小

在程序GP2206的基础上，若改变窗口的值（见140句），则得到图2.18。这是因为在视见区相同的情况下，窗口加大，图形变小。

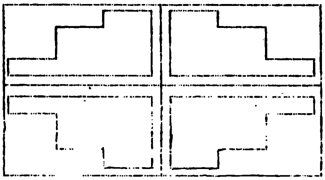


图 2.17　窗口：W1＝0：W2＝50：W3＝0：W4＝25

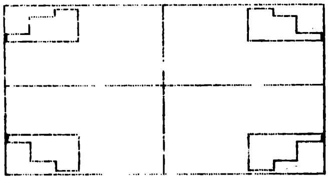


图 2.18　窗口：W1＝0：W2＝100：W3＝0：W4＝50

从上面两张可以看到，窗口值加一倍，图形成比例缩小一倍。

三、视见区扩大，图形变大

① 程序（GP2207）

```
130   TEXT : HOME
140   W1 = 0:W2 = 100:W3 = 0:W4 = 50

150   HGR2 : HCOLOR= 3
155   READ V1,V2,V3,V4: GOSUB 60000

160   GOSUB 63500: GOSUB 500
180   READ V1,V2,V3,V4: GOSUB 60000

190   GOSUB 63500: GOSUB 500
200   READ V1,V2,V3,V4: GOSUB 60000

210   GOSUB 63500: GOSUB 500
220   READ V1,V2,V3,V4: GOSUB 60000

230   GOSUB 63500: GOSUB 500
240   PRINT  CHR$ (7): END
300   DATA 0,139,0,95
310   DATA 279,139,0,95
320   DATA 0,139,191,95
```

```
340  DATA 139.5,209,90.5,47.5
350  DATA 279,209.5,90.5,47.5
360  DATA 139.5,209,47.5,0
370  DATA 279,209.5,90.5,0
```

② 运行结果（见图2.20）

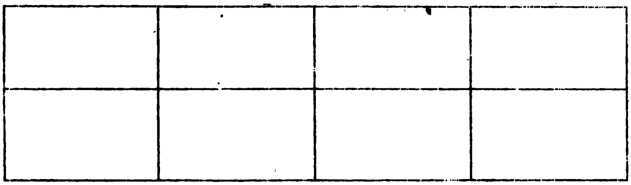


图 2.20

五、8个视见区图象镜像复制

程序GP 2209是在程序GP 2208的基础上，加上了图形数据，画出的8个对称图形（见图2.21）。

① 源程序（见GP2209）

```
130  TEXT : HOME
140 W1 = 0:W2 = 100:W3 = 0:W4 = 50

150  HGR2 : HCOLOR= 3
155  READ V1,V2,V3,V4: GOSUB 60000
     : GOSUB 400
160  GOSUB 63500
180  READ V1,V2,V3,V4: GOSUB 60000
     : GOSUB 400
190  GOSUB 63500
200  READ V1,V2,V3,V4: GOSUB 60000
     : GOSUB 400
210  GOSUB 63500
```

```
220  READ V1,V2,V3,V4: GOSUB 60000
     : GOSUB 400
230  GOSUB 63500
232  READ V1,V2,V3,V4: GOSUB 60000
     : GOSUB 400: GOSUB 63500
234  READ V1,V2,V3,V4: GOSUB 60000
     : GOSUB 400: GOSUB 63500
236  READ V1,V2,V3,V4: GOSUB 60000
     : GOSUB 400: GOSUB 63500
238  READ V1,V2,V3,V4: GOSUB 60000
     : GOSUB 400: GOSUB 63500
240  PRINT  CHR$ (7): END
300  DATA 0,69.5,0,47.5
310  DATA 139.5,69.5,0,47.5
320  DATA 0,69.5,90.5,47.5
330  DATA 139.5,69.5,90.5,47.5
370  DATA 279,209.5,0,47.5
380  DATA 139.5,209,0,47.5
385  DATA 279,209.5,90.5,47.5
390  DATA 139.5,209,90.5,47.5
400 X = 2:Y = 22:W$ = "U"
405  GOSUB 62300:W$ = "D"
410 X = 47:Y = 22: GOSUB 62300
420 X = 47:Y = 2: GOSUB 62300
430 X = 32:Y = 2: GOSUB 62300
440 X = 32:Y = 7: GOSUB 62300
450 X = 17:Y = 7: GOSUB 62300
460 X = 17:Y = 17: GOSUB 62300
470 X = 2:Y = 17: GOSUB 62300
480 X = 2:Y = 22: GOSUB 62300
490  RETURN
```

② 运行结果（见图2.21）

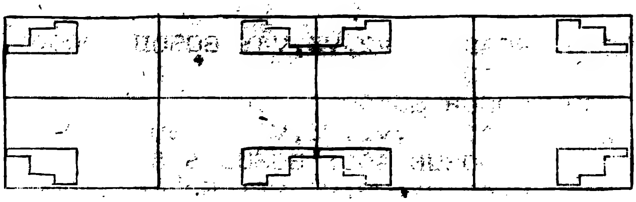


图 2.21

第三章　分页显示

在显示屏幕上，实现不同图象的分页显示，可使计算机辅助教学增加情趣，给读者留下深刻印象，从而提高学习效率。

图形显示的基本要求是：

· 能显示复杂图形

· 分页显示速度快

· 整体画面效果好

· 动态变化较突出

图形显示的基本原理是：

显示一幅图象，然后把它从屏幕上抹去，再显示另一幅图象，再予以抹去。

本章我们将介绍多幅画面的分页显示方法。

为叙述方便，我们首先介绍图形存贮、调用、图形搬家的技巧和屏幕软开关的作用，然后分节叙述多幅画面分页显示的技术。

第一节　图形的存贮和调用

屏幕上任何一幅图形都是由程序运行时产生的。因此，通常的方法的是将这些画图程序存入磁盘，以后再调入内存并运行它，图形便能再现。显然，这些图形的产生离不开画图程序的运行，如果画图程序十分复杂，图形信息十分丰富，

那么运行程序，显示画面的时间就会很长，这在分页快速显示图象时，就会很不方便。因此，我们应设法直接把图形存入磁盘。

方法也很简单，在每一幅图形完全画好并在屏幕上显示结束后，用下面任何一种命令，都可将图形信息存于磁盘中：

BSAVE 图形名，A＄2000，L＄2000

BSAVE 图形名，A＄4000，L＄2000

BSAVE 图形名，A＄6000，L＄2000

其中，图形名可不必与程序名同，以字母打头就可以。A＄后面的数字，表明图形页的起始地址，L＄后面的数字，表示图形区的长度。BSAVE命令，是把指定地址开始的指定长度的二进制信息以给定名字存在磁盘上，列目录时可以看到，它们是以B字头标志，占34个扇区。

应该指出的是：上述各种存贮格式，在调入内存时，必须指定将图形信息调入图形存贮区，是从那一个地址开始的。

将存于磁盘上的图形调出来，比较简单，可用命令：

BLOAD图形名

或者：

BLOAD 图形名，A＄2000 （调入第一页图形存贮区）

或者

BLOAD 图形名，A＄4000 （调入第二页图形存贮区）

或者

BLOAD图形名， A$6000 （在 $6000开始有贮图形）

值得注意的是， BLOAD命令只能把图形信息，调入相应的图形存贮区，它本身不包含显示功能。同时BLOAD图形名， A$2000的图形，只能在图形第一页看到，BLOAD图形名， A$4000的图形，只能在第二页看到；而BLOAD图形名， A$6000的图形，在第一页和第二页均不能看到，而必须将其搬至第一页或第二页，然后用 HGR或HGR2或软开关，才能显示出图形。

第二节 图形搬家

这有两种情况：

一是原来在内存的图形信息（例如存于第一页A$2000中），改搬到磁盘上另一个存贮地址单元（例如A$4000），可以用以下形式：

BSAVE图形名， A$4000， L$2000

这样，关机后，重新调入内存时，图形不是在第一页，而是在第二页。

二是原来在盘片中的图形信息（例如存于 A$6000，L$2000中），改搬到内存中另一个图形存贮区（如A$4000），可以用以下方法：

```
10 X$ ="4000<6000·7FFFM  N  D823G"
20FOR  I=1TO  LEN（X$）：POKE511+ I,
ASC（M1 D$（X$，I，1）+128：NEXT  I：
POKE 72,0
30CLAA－144
```

在此以前，应将盘片中的信息（原存在A＄6000一段地址的内容），调进内存。

当然，用DOS命令，会使这种搬家更为简捷，如：

PRINT CHR＄（4）“BLOAD 图形名，A＄4000”

或PRINT CHR＄（4）“BLOAD”图形名”，A＄4000

第三节　屏幕软开关

众所周知，用屏幕软开关可以实现屏幕工作方式的转换，在程序中只要访问这些软开关的地址，就能设置屏幕显示器的方式。如：

POKE	—16302，0	全屏幕图形显示
POKE	—16297，0	设定高分辨率方式
POKE	—16304，0	设定图形方式
POKE	—16300，0	选定高分辨第一页
POKE	—16299，0	选择高分辨第二页
POKE	—16301，0	图象文本显示方式

所要注意的是，执行HGR和HGR 2命令，都会把荧屏上的内容清除，这在需要保留原来的高分辨图象时十分不便。为了进行高分辨图形显示方式的转换，同时又不致于清除原来的高分辨图象，所以要用POKE语句来实现。

第四节　分页和连续显示的原理

高分辨图形有两个显示区，或者称有两个存贮区（通常又称第一页和第二页），每个存贮区占有8192个单元(即＄2000)，

两页存贮区地址安排如下：

高分辨图形第一页： $2000— $3FFF

高分辨图形第二页： $4000— $5FFF

为了达到分页显示或连续显示的目的，可以用一个存贮区来存放信息，另一个存贮区用于取出信息并显示，然后掉转过来，一个用于显示，另一个用于存贮，从而实现连续显示。

下面，我们通过若干实例，详细介绍多幅画面分页显示的实施方法。

第五节　单幅画面的显示

有一个LOGO图形（见图2.22），以SAVEPICT "HY.PICT↙存入磁盘，如何用BASIC程序将其显示并打印出来。

编制一个BASIC程序，见GP2301，然后按下述步骤执行操作：

- 引导DOS进入BASIC状态。
- 将显示打印程序GP2301调进内存。
- 放LOGO图形资料盘片进驱动器。
- 调被显示的LOGO图形进内存，如BLOAD HY.PICT↙。
- 运行程序GP2301。

程序GP2301清单。

```
10  POKE  - 16304,0
20  POKE  - 16297,0
30  POKE  - 16302,0
40  POKE  - 16300,0
```

```
45   FOR I = 1 TO 2500: NEXT I
50   TEXT : HOME
60   PR# 1
70   POKE 1913,1
80   PRINT  CHR$ (17)
90   PR# 0
100  END
```

LOGO图象，见图2.22。

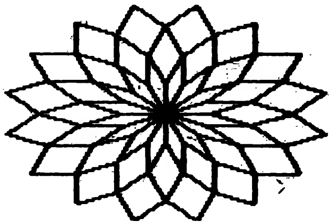


图 2.22

说明：由于 LOGO 图形都是存贮在高分辨率第一页的，所以在程序中安排了POKE －16300，0（置第一页），同时为了看清图象的全过程，安置了一个延迟，即FOR I＝1 TO2500：NEXT。

如果仅仅显示图形，也可用程序GP 2302来实现。

程序GP 2302清单。

```
10   HGR2 : HCOLOR= 7
15   INPUT A$
20   PRINT  CHR$ (4);"BLOAD",A$
30   FOR I = 1 TO 2000: NEXT
40   HOME : TEXT
50   GOTO 10
```

程序GP 2302具有更大的灵活性，它可以根据输入的图形名，任意显示并可以不间断进行。

对于显示第二页的图形，程序GP 2301中的POKE －16300，0应改为POKE －16299,0;而程序 GP 2302 中的

HGR应改为HGR 2。

为实用的方便，可以编制更简单的显示程序，它将调进内存的第一页图形显示出来：

如] BLOAD X1 ↙

] RUN GP 2303 ↙

程序GP 2303清单。

```
10  POKE  - 16304,0
20  POKE  - 16297,0
30  POKE  - 16302,0
40  POKE  - 16300,0
50  FOR I = 1 TO 2000: NEXT I
60  TEXT : HOME
70  END
```

X1为假定的一个图形名。从这里可以进一步体会到，BLOAD命令不具有显示图形的功能，而必须加GP 2303的显示程序，才能将图形显示出来。

第六节 两幅画面的分页显示

用置换屏幕软开关方法，很容实现两幅画面的分页显示，见程序GP 2304。

```
10  REM GP2304
20  HOME
30 D$ =  CHR$ (4)
40  PRINT D$"BLOAD PICT1,A$2000"
50  PRINT D$"BLOAD PICT2,A$4000"
60  POKE 49232,0: POKE 49239,0
70  POKE 49234,0: POKE 49236,0
80  FOR J = 1 TO 2000: NEXT J
90  POKE 49237,0
```

```
100   FOR P = 1 TO 2000: NEXT P
110   GOTO 70
```

程序GP 2304中PICT 1和PICT 2为两幅画面的图形名称，可根据实际图形名称选定。80句和100句可以控制画面在屏幕上的停留时间，也可由实际需要灵活改变。如果PICT 1 和PICT 2 是表示的同一种图形的不同位置，则程序GP 2304还可以将该图形以动态方式显示出来。

第七节　三幅画面的分页显示

运行程序GP 2305，可以分页显示三幅图形。

```
10 D$ = CHR$ (4)
20   PRINT D$;"BLOAD GMS8,A$2000"
25   PRINT D$;"BLOAD P4,A$4000"
30   PRINT D$;"BLOAD GMS10,A$6000"
40   POKE 49232,0
50   POKE 49239,0
60   POKE 49234,0
70   POKE 49236,0
80   GET A$
90   POKE 49237,0
100   GET A$
110 X$ = "2000<6000.7FFFM N D823G"

120   FOR I = 1 TO LEN (X$): POKE
      511 + I, ASC ( MID$ (X$,I,1))
       + 128: NEXT I: POKE 72,0
130   CALL - 144
140   POKE 49236,0
150   GET A$
210   HOME : TEXT : HTAB 15: VTAB 1
      0: INVERSE : PRINT "GOOD BYE!
```

```
        ": NORMAL
220  END
```

程序 GP 2305还有这样一个特点：可以用按任意键控制画面在屏幕上的停留时间。

10—30句将三幅图形资料一次调进内存，40—60句置全屏幕高分辨率图形状态，70句显示第一页图形，按任意键(80句)显示第二页图形（90句)，再按任意键将第三页图形搬至第一页图形显示区（110—130句)，然在在第一页显示(140句)，最后按任意键清屏，然后反相显示GOOD BYE！字样结束。

第八节　三幅画面分页显示的自动控制

程序 GP 2305虽然能够用手动控制三幅画面，也能控制在屏幕上显示时间，但不能改变顺序，任意显示任一幅画面。为此，再介绍一种显示程序。

程序 GP 2306能够自动显示三幅图形，也能反复和任意显示任一幅画面，输入－1↙，结束程序运行。

假设磁盘中有N幅图形，它们都存放在 \$4000—\$5 FFF单元中，程序运行后可根据提示，任意选择磁盘中的图形，程序能自动搬家放置在第四页(\$8000—\$9 FFF)、第三页(\$6000—\$7 FFF)、第二页（\$4000—\$5 FFF）中，然后，程序又会根据输入的X值（X＝1,2,3)，自动显示任一幅画面，它们都利用第一页显示页面(\$2000—\$3 FFF)作为公共显示区。

由于中华学习机内存的限制，程序 GP 2306运行前，必须将 DOS 搬家。例如程序运行前，首先运行DOS　MOVER

程序,即BRUN DOS MOVER↙,然后再运行 GP 2306 程序。

程序GP 2306清单。

```
5   REM   AUTU DISPLAY
10 D$ =  CHR$ (4)
25  PRINT : PRINT
30  INPUT "NETER PICTURE FILE:A1$
     ";A1$
40  PRINT D$;"BLOAD",A1$
50 K1$ = "8000<4000.5FFFM N D823G"

52  GOSUB 1010: CALL  - 144
55  PRINT
60  INPUT "NETER PICTURE FILE:A2$
     ";A2$
70  PRINT D$;"BLOAD",A2$
80 K2$ = "6000<4000.5FFFM N D823G"

82  GOSUB 2010: CALL  - 144
85  PRINT
90  INPUT "NETER PICTURE FILE:A3$
     ";A3$
100  PRINT D$;"BLOAD",A3$
110  POKE 49232,0: POKE 49239,0: POKE
     49234,0
125  GOSUB 1000
175  CALL  - 144: POKE 49236,0: FOR
     I = 1 TO 500: NEXT I
176  GOTO 280
178  GOSUB 2000
225  CALL  - 144: POKE 49236,0: FOR
     I = 1 TO 500: NEXT I
226  GOTO 280
228  GOSUB 3000
275  CALL  - 144: POKE 49236,0: FOR
     I = 1 TO 500: NEXT I
```

```
278  PRINT
279  POKE 49232,0: POKE 49239,0: POKE
     49234,0
280  INPUT "ENTER PICTURE NUMBER "
     ;X
282  IF X =  - 1 THEN  TEXT : HOME
     : END
290  ON X GOTO 125,178,228
300  GOTO 280
1000 K1$ = "2000<4000.5FFFM N D823
     G"
1010  FOR I = 1 TO  LEN (K1$): POKE
     511 + I, ASC ( MID$ (K1$,I,1)
     ) + 128: NEXT I: POKE 72,0
1020  RETURN
2000 K2$ = "2000<6000.7FFFM N D823
     G"
2010  FOR I = 1 TO  LEN (K2$): POKE
     511 + I, ASC ( MID$ (K2$,I,1)
     ) + 128: NEXT I: POKE 72,0
2020  RETURN
3000 K3$ = "2000<8000.9FFFM N D823
     G"
3010  FOR I = 1 TO  LEN (K3$): POKE
     511 + I, ASC ( MID$ (K3$,I,1)
     ) + 128: NEXT I: POKE 72,0
3020  RETURN
```

第九节　四幅画面的分页显示

设有四幅画面，它们在磁盘中存贮情况为：

GMS 8：　$2000—$3 FFF

P 2　　　$4000—$5 FFF

GM10 ：　$6000—$7 FFF

MAP： $8000－$9FFF

欲将它们分页显示，其基本方法和三幅画面分页显示程序十分类似，主要不同点是用二页显示区作公用显示区，同上面三幅画面分页显示一样，也要首先将DOS搬家。

四页分页显示程序见GP 2307，运行该程序后，四幅画面分别调入内存。然后先在第一页显示GMS 8图形(70句)，稍等片刻，显示在第二页显示P 2图形（90句)，经100句延迟后，将原来存在内存第二页页的图形搬至第一页（110—130句)，执行140句在第一页显示GMS 10图形，再经150句延迟片刻，将原来第四页的图形搬至第二页（160—180句)，执行1[illegible]句在第二页显示MAP图形，最后清屏结束。

程序GP 2307清单。

```
10 D$ =  CHR$ (4)
20  PRINT D$;"BLOAD GMS8"
25  PRINT D$;"BLOAD P2"
30  PRINT D$;"BLOAD GMS10"
35  PRINT D$;"BLOAD MAP"
40  POKE 49232,0
50  POKE 49239,0
60  POKE 49234,0
70  POKE 49236,0
80  FOR I = 1 TO 1500: NEXT
90  POKE 49237,0
100  FOR I = 1 TO 1500: NEXT
110 X$ = "2000<6000.7FFFM N D823G"

120  FOR I = 1 TO  LEN (X$): POKE
     511 + I, ASC ( MID$ (X$,I,1))
      + 128: NEXT I: POKE 72,0
130  CALL  - 144
140  POKE 49236,0
```

```
150  FOR I = 1 TO 2500: NEXT
160 Y$ = "4000<8000.9FFFM N D823G"

170  FOR I = 1 TO  LEN (Y$): POKE
     511 + I, ASC ( MID$ (Y$,I,1))
      + 128: NEXT I: POKE 72,0
180  CALL  - 144
190  TUKE 49237,0
200  FOR I = 1 TO 2500: NEXT
210  TEXT : HOME
220  END
```

为使四幅画面显示更加灵活方便，可用程序 GP2308，其特点是：

·磁盘中图形可以存放在任意区（$2000，$4000，$6000，$8000均可）；

·可以随意调磁存中任意 4 幅画面，一经调入内存，程序会自动分别放在相应存贮区；

·有防止按键出错的处理（108句）；

·有自动停机的功能（106句）；

·有自由选择任一幅画面的方法；

·全部画面只用了第二页作为公共显示区。

运行 GP2308程序前，应将 DOS 搬家。

显然，本程序编制简捷，使用灵活，技巧性强。特别是150句的安排。如果150句中 CALL－144 放在 200 句 RETURN前，程序将发生错误，读者可以上机试验，并分析原因。

程序 GP2308清单。

```
5  REM GP2308
10 D$ =  CHR$ (4): HOME : PRINT D$
     "CATALOG"
15  PRINT
20  INPUT "PICT1 ";A$
30  INPUT "PICT2 ";B$
40  INPUT "PICT3 ";C$
50  INPUT "PICT4 ";E$
60  PRINT D$"BLOAD"A$",A$2000
70  PRINT D$"BLOAD"B$",A$4000
80  PRINT D$"BLOAD"C$",A$6000
90  PRINT D$"BLOAD"E$",A$8000
100 K$ = "1000<4000.4FFFM N A000<5
     000.5FFFM N D823G": GOSUB 200
     : CALL  - 144: HGR2
104  GET A$
106  IF A$ = " " THEN  TEXT : HOME
     : VTAB 10: HTAB 20: FLASH
107  PRINT "END!": NORMAL : END
108  IF A$ < "1" OR A$ > "4" THEN
     104
110  IF A$ = "1" THEN K$ = "4000<2
     000.3FFFM N D823G": GOSUB 200

120  IF A$ = "2" THEN K$ = "4000<6
     000.7FFFM N D823G": GOSUB 200

130  IF A$ = "3" THEN K$ = "4000<8
     000.9FFFM N D823G": GOSUB 200

140  IF A$ = "4" THEN K$ = "4000<1
     000.1FFFM N 5000<A000.AFFFM N
      D823G": GOSUB 200
150  CALL  - 144:F = FRE (0): GOTO
     104
200  FOR I = 1 TO  LEN (K$): POKE
     511 + I, ASC ( MID$ (K$,I,1))
```

```
    + 128: NEXT I: POKE 72,0
210 RETURN
```

第十节 八幅画面的分页显示

GP 2309是一个可以分页显示八幅画面的程序，其中20句调八幅图形压缩存贮的资料(文件名为M-T-TXZL8)，30句调压缩存贮程序（文件名为GP 2310)。运行GP 2309程序后，可以从键盘输入1—8各个数字，从而手动控制显示八幅不同的画面。

整个程序采用总控的方式（见90句)，从而可以任意显示八幅画面中的任一个画面。

100句中POKE 230，32是指第一页显示区，$0和$1号单元中的A、B值，是指压缩图形存放的地址，而110句CALL 4099是调用还原图形，80句是设置第一页高分辨图形全屏幕显示。

整个运行过程，由于采用手动控制，可以自由控制显示时间，也可方便地自由选取任一幅画面显示，停止运行按数字“0”。

程序GP 2309清单。

```
10 D$ = CHR$ (4): HGR
20  PRINT D$"BLOAD M-T-TXZL8"
30  PRINT D$"BLOAD GP2310"
35 N = 1: GOTO 80
40 X = PEEK ( - 16384)
50  IF X = 176 THEN POKE - 16368
     ,0: HOME : TEXT : END
60  IF X < 176 OR X   184 THEN 40
70 N = X - 176: POKE - 16368,0
```

```
80  HCOLOR= 3
85  POKE  - 16302,0: POKE  - 16300
     ,0: POKE  - 16304,0: POKE  -
     16297,0
90  ON N GOTO 120,130,140,150,160,
     170,180,190
100  POKE 0,A: POKE 1,B: POKE 230,
     32
110  CALL 4099: GOTO 40
120 A = 0:B = 64: GOTO 100
130 A = 16:B = 73: GOTO 100
140 A = 96:B = 84: GOTO 100
150 A = 0:B = 96: GOTO 100
160 A = 0:B = 104: GOTO 100
170 A = 0:B = 110: GOTO 100
180 A = 0:B = 123: GOTO 100
190 A = 0:B = 134: GOTO 100
```

程序GP2310清单。

```
1000- 4C 7F 10 A5 E6 09 04 85
1008- 06 A2 01 86 04 A0 00 84
1010- 02 84 05 84 08 B1 00 D0
1018- 18 E6 00 D0 02 E6 01 B1
1020- 00 85 08 E6 00 D0 02 E6
1028- 01 B1 00 85 07 A5 07 C6
1030- 08 A4 02 91 05 E8 E8 E0
1038- BF 90 12 E6 02 A4 02 C0
1040- 28 90 08 C6 04 30 15 A0
1048- 00 84 02 A6 04 20 5D 10
1050- A4 08 D0 D9 E6 00 D0 BD
1058- E6 01 D0 B9 60 8A 29 C0
1060- 85 05 4A 4A 05 05 85 05
1068- 8A 85 06 0A 0A 0A 26 06
1070- 0A 26 06 0A 66 05 A5 06
1078- 29 1F 05 E6 85 06 60 A0
```

```
1080- 01 84 04 84 03 88 84 02
1088- A5 E6 09 04 85 06 84 05
1090- B1 05 D0 02 09 80 A2 01
1098- 86 08 85 07 A4 02 A6 03
10A0- E8 E8 E0 BF 90 0F C8 C0
10A8- 28 90 06 C6 04 30 1B A0
10B0- 00 84 02 A6 04 86 03 20
10B8- 5D 10 B1 05 D0 02 09 80
10C0- C5 07 D0 06 E6 08 D0 D4
10C8- C6 08 48 A0 00 A6 08 F0
10D0- 0E E0 04 B0 0A A5 07 20
10D8- F2 10 CA D0 FA F0 0D 98
10E0- 20 F2 10 8A 20 F2 10 A5
10E8- 07 20 F2 10 68 24 04 10
10F0- A5 60 91 00 E6 00 D0 02
10F8- E6 01 60
```

如果欲自动显示八幅图形，可运行程序GP 2311，每幅画面在屏幕上的显示时间，可改变100句循环终值。

程序GP 2311清单。

```
10  REM GP2311
20 D$ =  CHR$ (4): HGR
30  PRINT D$"BLOAD M-T-TXZL8"
40  PRINT D$"BLOAD GP2310"
50  FOR N = 1 TO 8
60  HCOLOR= 3
65  POKE  - 16302,0: POKE  - 16300
     ,0: POKE  - 16304,0: POKE  -
     16297,0
70  ON N GOTO 120,130,140,150,160,
     170,180,190
80  POKE 0,A: POKE 1,B: POKE 230,3
     2
```

```
90  CALL 4099
100  FOR I = 1 TO 2000: NEXT I
110  GOTO 200
120 A = 0:B = 64: GOTO 80
130 A = 16:B = 73: GOTO 80
140 A = 96:B = 84: GOTO 80
150 A = 0:B = 96: GOTO 80
160 A = 0:B = 104: GOTO 80
170 A = 0:B = 110: GOTO 80
180 A = 0:B = 123: GOTO 80
190 A = 0:B = 134: GOTO 80
200  NEXT N
210  TEXT : HOME : END
```

第十一节 翻页显示

GP 2312是一个用机器语言编写的翻页显示程序，它可以使送入内存的第一页画面，从左下角斜着翻起，并露出第二页画面，直到第一页画面完全被第二页画面所取代。全部过程就像翻小人书一样，使页面转换更有一番情趣。

GP 2313是一个用 BASIC 语言编写的程序，运行后可以看到上述两个画面的翻页显示过程。在GP 2313中，首先分别调两幅画面 C1和 C2进内存，然后调机器语言程序GP 2312进内存，用 HGR : POKE −16302，0 设置高分辨第一页为全屏幕显示状态，再执行 CALL 768 命令，就可看到翻页显示的过程，经过延迟处理，再不断循环，结果周而复始地看到第二页画面 C 2 的翻页显示，而要停止，按 CTRL-RESET↙。

程序GP 2312清单。

```
0300- 4C 29 03 A5 11 0A 0A 29
0308- 1C 85 15 A5 11 6A 6A 6A
0310- 6A 29 03 05 15 09 20 85
0318- 15 A5 11 6A 29 E0 85 14
0320- 6A 6A 29 18 05 14 85 14
0328- 60 A0 00 84 A2 84 02 A2
0330- BF 86 11 A2 03 20 03 03
0338- A5 14 85 00 A5 15 69 20
0340- 85 01 B1 00 91 14 E8 C6
0348- 11 E0 05 90 E8 88 A2 00
0350- C0 FF 90 E1 E6 02 A4 02
0358- A9 FF C5 11 D0 D1 A2 BF
0360- 86 11 86 02 A2 03 A5 02
0368- 38 E9 05 18 86 03 65 03
0370- 85 02 A0 27 20 03 03 A5
0378- 14 85 00 A5 15 69 20 85
0380- 01 B1 00 91 14 E8 C6 11
0388- E0 05 90 E8 88 A5 11 C9
0390- FF A2 00 90 DF A6 02 86
0398- 11 A2 00 A5 02 C9 FF D0
03A0- C5 60 11 A2 00 D0 C0 60
```

程序GP 2313清单。

```
10  REM GP2313
20 D$ =  CHR$ (4)
30  PRINT D$"BLOAD C1,A$2000"
40  PRINT D$"BLOAD C2,A$4000"
50  PRINT D$"BLOADGP2312,A$300"
60  HGR : POKE  - 16302,0
70  CALL 768
80  FOR I = 1 TO 3000: NEXT
90  GOTO 60
```

RUN GP 2313↙后，看到图形C 2的翻页显示，见图2.23。

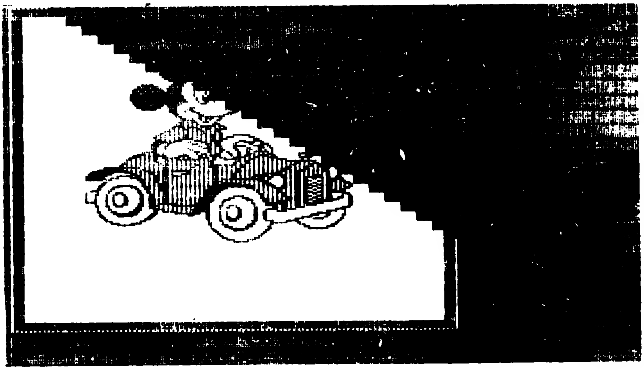


图 2.23

图形C2的完整画面，见图2.24。

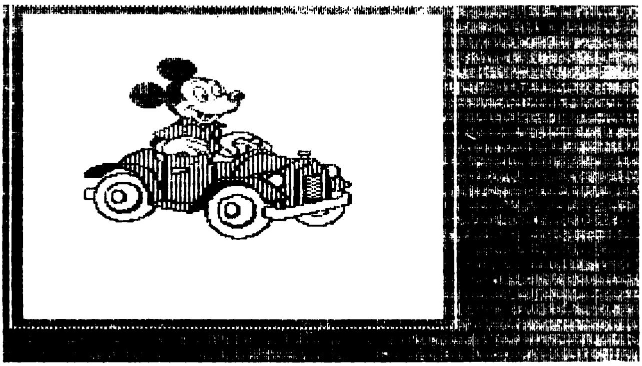


图 2.24

如果将程序GP 2313作些改动，即利用搬家的方法，可以看到两页画面轮流翻页显示，机器语言程序仍用GP 2312，而BASIC程序改用GP 2314。

程序GP 2314清单。

```
5  ONERR  GOTO 160
10  REM  GP2314
20 D$ =  CHR$ (4)
30  PRINT D$"BLOAD C1,A$2000"
40  PRINT D$"BLOAD C2,A$4000"
50  PRINT D$"BLOAD GP2312,A$300"
60  POKE  - 16297,0: POKE  - 16302
     ,0: POKE  - 16304,0: POKE  -
     16300,0
70  FOR I = 1 TO 3000: NEXT
80 K$ = "6000<2000.3FFFM N D823G"
90  FOR I = 1 TO  LEN (K$): POKE 5
     11 + I, ASC ( MID$ (K$,I,1)) +
     128: NEXT I: POKE 72,0
100  CALL  - 144
110  CALL 768
120 K$ = "4000<6000.7FFFM N D823G"

130  FOR I = 1 TO  LEN (K$): POKE
     511 + I, ASC ( MID$ (K$,I,1))
      + 128: NEXT I: POKE 72,0
140  CALL  - 144
150  GOTO 80
160  TEXT : HOME : END
```

图形C 1的原图象，见图2.25。

图形C 2的原图象，见图2.26。

程序GP 2315可以显示三幅画面的翻页显示，其显示顺序为C 1，C 2，C 3，C 2，C 1。

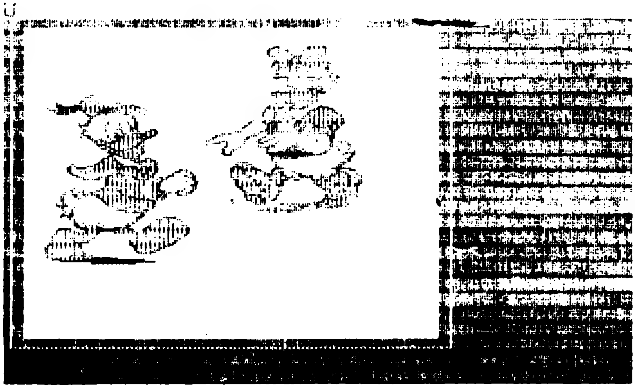


图 2.25

```
*
]POKE1913,33
```

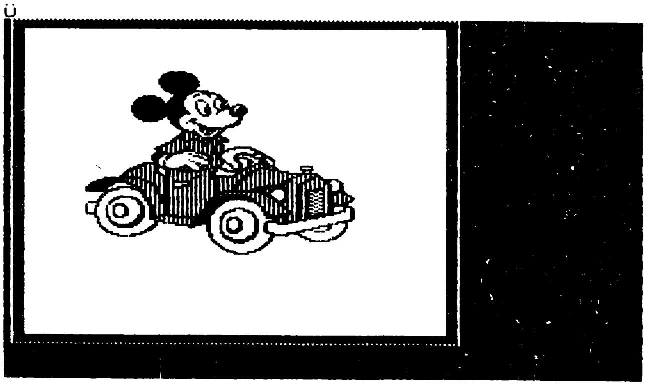


图 2.26

```
5  ONERR  GOTO 300
10  REM GP2315
20 D$ =  CHR$ (4)
30  PRINT D$"BLOAD C1,A$2000"
40  PRINT D$"BLOAD C2,A$4000"
45  PRINT D$"BLOAD C3,A$6000"
50  PRINT D$"BLOAD GP2312,A$300"
60  POKE  - 16297,0: POKE  - 16302
     ,0: POKE  - 16304,0: POKE  -
     16300,0
70  FOR I = 1 TO 3000: NEXT I
80 K$ = "1000<2000.2FFFM N 8000<30
     00.3FFFM N D823G"
90  FOR I = 1 TO  LEN (K$): POKE 5
     11 + I, ASC ( MID$ (K$,I,1)) +
     128: NEXT I: POKE 72,0
100  CALL  - 144
110  CALL 768
112 K$ = "2000<6000.7FFFM N D823G"

114  FOR I = 1 TO  LEN (K$): POKE
     511 + I, ASC ( MID$ (K$,I,1))
      + 128: NEXT I: POKE 72,0
116  CALL  - 144
118  CALL 768
120 K$ = "4000<2000.3FFFM N D823G"

130  FOR I = 1 TO  LEN (K$): POKE
     511 + I, ASC ( MID$ (K$,I,1))
      + 128: NEXT I: POKE 72,0
140  CALL  - 144
150  CALL 768
160 K$ = "4000<1000.1FFFM N 5000<8
     000.8FFFM N D823G"
170  FOR I = 1 TO  LEN (K$): POKE
     511 + I, ASC ( MID$ (K$,I,1))
```

```
      + 128: NEXT I: POKE 72,0
180  CALL  - 144
200  CALL 768
210 K$ = "4000<2000.3FFFM N D823G"

220  FOR I = 1 TO  LEN (K$): POKE
     511 + I, ASC ( MID$ (K$,I,1))
      + 128: NEXT I: POKE 72,0
230  CALL  - 144
232  CALL 768
300  TEXT : HOME : END
```

图形C 3的原图象，见图2.27。

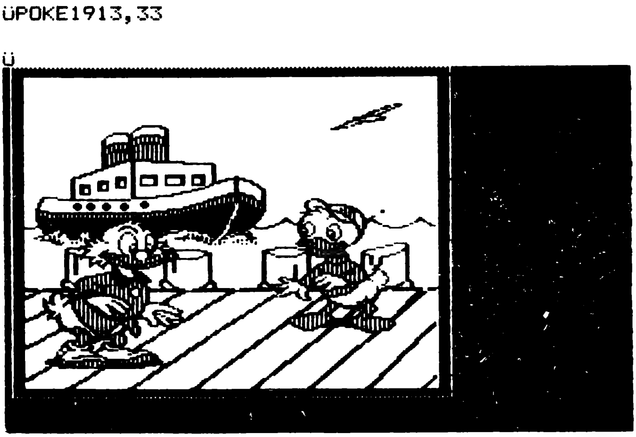


图 2.27

如果改变程序的编制方法，还可以有更多的翻页显示方

式，图形画面也可以取多页，但要采取一定的措施。如图形压缩存贮、利用扩充卡、语言卡、 DOS 搬家等技术，使内存中可以容纳更多的画面。

第四章 图 象 处 理

本章介绍最简单的 4 种图象处理技术，即将调进内存的任意一幅图象，进行左右易位、水平镜象、垂直压缩、成倍翻番处理，从而使图形画面更加生动活泼、丰富多采。

上述 4 种处理技术，都采用机器语言实现，而为了能对以上处理技巧连贯起来操作，编制一个BASIC 程序，将是十分方便的，在BASIC程序中调用机器语言子程序。

第一节 左 右 易 位

GP 2401是一个图形左右易位的处理子程序，它可以将高分辨率第二页的图形，进行左右移位后在高分辨率第一页上显示出来，使之耳目一新，生动有趣。

首先，将欲处理的图形（例如P 2）调入第二页： BLOAD P 2，A$ 4000↙。

然后，执行GP 2401机器语言图形处理子程序：] CALL 768↙（或＊300G↙）。

最后，在高分辨率第一页画面上出现左右翻身了的图形。

为了看清原来在第二页上未处理的图象，以及在第一页上处理后的图象，可以用下面程序段，实现分页显示：

```
300  FOR K = 1 TO 5
310  POKE  - 16297,0: POKE  - 1630
```

```
      2,0: POKE  - 16304,0: POKE  -
      16300,0
320   FOR I = 1 TO 2000: NEXT I
330   POKE  - 16299,0
340   FOR I = 1 TO 2000: NEXT I
350   NEXT K
360   TEXT : HOME : END
```

第二页上的原图形，见图2.28(a)。

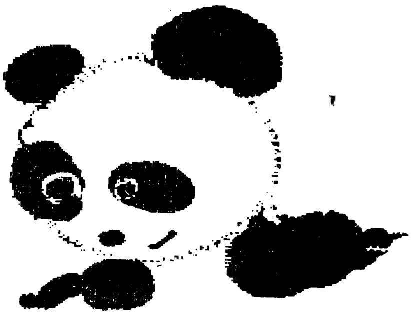


图 2.28(a)

第一页上处理后的图形，见图2.28(b)。



图 2.28(b)

程序GP 2401清单。

```
0300- A2 00 8A 48 20 11 F4 68
0308- AA A5 26 9D 00 60 A5 27
0310- 9D 00 61 E8 E0 C0 D0 EA
0318- A9 27 85 0A A0 00 A2 00
0320- BD 00 60 85 06 85 08 BD
0328- 00 61 85 07 69 20 85 09
0330- B1 08 85 FF 06 FF 08 A9
0338- 00 85 0B 06 FF 66 FE E6
0340- 0B A5 0B C9 07 D0 F4 28
0348- 66 FE 98 48 A4 0A A5 FE
0350- 91 06 68 A8 E8 E0 C0 D0
0358- C7 C6 0A C8 C0 28 D0 BE
0360- 60
```

应该指出的是，在上述机器语言执行后，有时会出现不能左右移位的情况，此时只要检查一下＄E 6单元的值。可以看出＄E 6单元的值不再是20，即发生错误。应重置即：＊E 6: 20，或 在 BASIC 状态下执行 POKE 230,32↙ 即可。

程序 GP 2402也是一个左右易位（或者左右翻身）的机器语言子程序，同样可以对高分辨图形进行左右变换。

首先在监控状态下健入GP 2402机器语言，返回 BASIC 状态后，调一个高分辨图形进内存，如BLOAD P2， A＄2000↙，然后键入 CALL 768↙，则在高分辨第一页上看到的是翻了身的图象。

当然不要忘记为了能看到图象，必须用软开关设置，即：

POKE －16304，0：POKE －16297，0：

POKE －16302，0：POKE －16300，0

程序 GP 2402存贮在＄300—＄353单元中。

程序 GP 2402清单。

```
0300- A9 00 A2 00 A0 00 85 EC
0308- 20 11 F4 A0 00 B1 26 20
0310- 3E 03 A5 EE 85 EF 98 AA
0318- 8E 1F 03 A9 27 38 E9 13
0320- A8 B1 26 20 3E 03 A5 EF
0328- 91 26 8A A8 A5 EE 91 26
0330- C8 C0 14 D0 D8 E6 EC A5
0338- EC C9 C0 D0 C5 60 86 ED
0340- 2A 26 EE 6A A2 07 6A 26
0348- EE CA E0 00 F0 03 4C 46
0350- 03 A6 ED 60
```

第二节 水平镜象

GP 2403是一个图形水平镜象处理子程序，它可以将第二页高分辨率的图形，变成对称显示的双幅图象，使之更富有艺术性。

首先，将要处理的图形（1个熊猫）置入高分辨率第二页，然后再执行GP 2403程序（CALL 768↙），就可以得到图2.29的图形（同本章第一节的方法，要显示图形，必须加适当的显示程序）。

这种整页对称复制图形的技术，在设计对称题图时，特别有用。当然，上述处理后的图象，水平压缩了，显得有些“苗条”。

程序GP 2403清单。

```
0300-  A2 00 8A 48 20 11 F4 68
0308-  AA A5 26 9D 00 60 A5 27
0310-  9D 00 61 E8 E0 C0 D0 EA
0318-  A2 00 A0 00 84 FD 84 FE
0320-  84 08 BD 00 60 85 06 85
0328-  F9 BD 00 61 85 FA 18 69
0330-  20 85 07 B1 06 85 0B C8
0338-  B1 06 85 0A A9 00 85 09
0340-  E6 09 46 0B 66 FD 46 0B
0348-  66 FE A5 09 C9 03 D0 F0
0350-  A9 00 85 09 46 0B 66 FD
0358-  46 0A 66 FE E6 09 46 0A
0360-  66 FD 46 0A 66 FE A5 09
0368-  C9 03 D0 F0 66 FD 66 FE
0370-  98 48 A5 FD 05 FE A4 08
0378-  91 F9 0A 08 0A 66 FF 0A
0380-  66 FF 0A 66 FF 0A 66 FF
0388-  0A 66 FF 0A 66 FF 0A 66
0390-  FF 28 66 FF A9 27 38 E5
0398-  08 A8 A5 FF 91 F9 68 A8
03A0-  C8 E6 08 C0 28 D0 8C E8
03A8-  E0 C0 D0 01 60 4C 1A 03
```

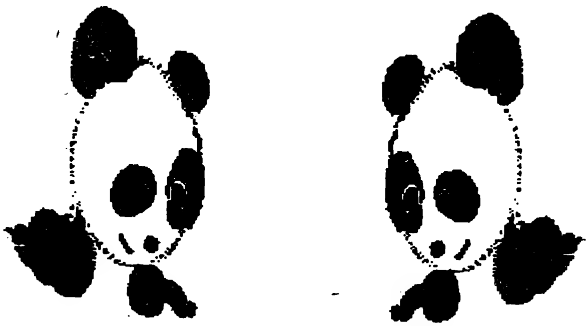


图 2.29

第三节 垂直压缩

GP 2404是一个图形垂直压缩处理子程序，它可以将第二页高分辨的图形，经过垂直压缩加工变换，变成缩小了的图象，使之又有一番情趣。

操作方法是，首先将要处理的图形（例如水平镜像处理的两个细长图形，见图2.29）。移至＄4000，然后执行CALL 768↙（程序GP 2404应先调入内存），即可得到如图2.30的图象。



图 2.30

程序 GP 2404清单。

```
0300- A2 00 8A 48 20 11 F4 68
0308- AA A5 26 9D 00 60 A5 27
0310- 9D 00 61 E8 E0 C0 D0 EA
0318- A9 00 85 0A A2 00 A0 00
0320- BD 00 60 85 06 BD 00 61
0328- 18 69 20 85 07 E8 BD 00
0330- 60 85 08 BD 00 61 18 69
0338- 20 85 09 B1 06 11 08 85
0340- 0B 8A 48 A6 0A BD 00 60
0348- 85 FE BD 00 61 85 FF A5
0350- 0B 91 FE 68 AA C8 C0 28
0358- D0 E1 E8 E6 0A E0 C0 D0
0360- BD 60
```

第四节 成倍翻番

如果将一页画面上的两幅图象，经上述水平镜像处理后，可以得到4幅图象，同理将4幅图象（在一页上），再经一次水平镜像处理，则可得到8幅图象，这就是图象成倍翻番。下面是一些实例。

原第二页上的两个熊猫，经GP 2403程序处理后，变成4个熊猫，见图2.31(a)。

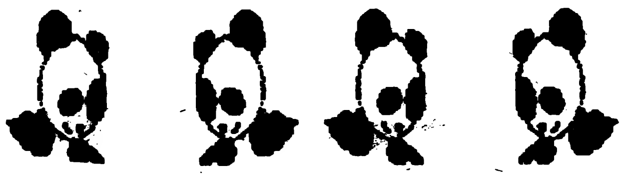


图 2.31(a)

原第二页上的4个熊猫，经GP 2404程序处理后，垂直方向上压缩了，见图2.31(b)。

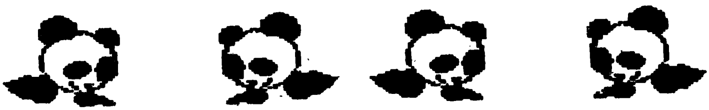


图 2.31(b)

对GP 2401，GP 2403，GP 2404 3个机器语言子程序，连贯起来的操作，请见GP 2405程序，这样可以简化操作手续，由程序自动控制完成，并可以连续看到屏幕上各种变化的图象，栩栩如生，引人入胜。

程序 GP 2405清单。

```
10 D$ =  CHR$ (4)
20  PRINT D$"BLOAD P2,A$4000"
30  PRINT D$"BLOAD GP2401"
40  POKE 230,32
50  CALL 768
60  GOSUB 300
70  GOSUB 360
80  CALL  - 144
90  PRINT D$"BLOAD GP2403"
100  CALL 768
110  GOSUB 300
120  GOSUB 360
130  CALL  - 144
140  PRINT D$"BLOAD GP2404"
150  HGR : POKE  - 16302,0
160  CALL 768
170  POKE  - 16300,0
180  FOR I = 1 TO 2000: NEXT I
190  TEXT : HOME : END
300  FOR K = 1 TO 3
310  POKE  - 16297,0: POKE  - 1630
     2,0: POKE  - 16304,0: POKE  -
     16300,0
320  FOR I = 1 TO 2000: NEXT I
330  POKE  - 16299,0
340  FOR I = 1 TO 2000: NEXT I
350  NEXT K: RETURN
360 X$ = "4000<2000.3FFFM N D823G"

370  FOR I = 1 TO  LEN (X$): POKE
     511 + I, ASC ( MID$ (X$,I,1))
      + 128: NEXT I: POKE 72,0
380  RETURN
```

第五章　合并显示

利用中华学习机或APPLEⅡ的高分辨作图时，有时需要将两页或更多页画面合并显示；或者，在绘制一幅比较复杂的图画时，为了尽快完成，可以将这幅画分成几个部分，分头由几个人同时绘制，然后再合而为一，这就需要一个图形合并程序。

图形合并程序，可以采用BASIC语言编制，但速度较慢；也可以采用机器语言编制，其执行速度可以大大提高。再巧妙地利用屏幕软开关，可以实现屏幕方式的快速转换，从而使图形画面更加生动活泼。

本章首先通过两幅画面的合并过程，介绍图形合并的基本方法和技巧，然后通过重新组织程序，采用分页显示技术，从而使画面更加多样，最后再介绍多幅画面合成的方法。

第一节　两幅画面的合并

设有两幅图形资料GMS8和P2，它们分别存放在磁盘片中，前者在$2000－$3FFF中，后者在$4000－$5FFF中，现在的任务是将这两幅画面合并在一张画面上，并让它们显示出来。

首先，在监控状态下，按入机器语言合并子程序GP2501，存盘备用（BSAVE　GP2501，A$300，L$26）。

```
0300- A9 00 85 00 A9 20 85 01
0308- A9 00 85 02 A9 40 85 03
0310- A0 00 B1 02 F0 02 91 00
0318- C8 D0 F7 E6 03 E6 01 A5
0320- 01 C9 40 90 ED 60
```

然后，在BASIC状态下，键入程序GP 2502，存盘备用（SAVE GP 2502)。

```
10 D$ =  CHR$ (4)
20  PRINT D$;"BLOAD GMS8"
30  PRINT D$;"BLOAD P2"
40  PRINT D$;"BRUN GP2501"
50  POKE  - 16297,0: POKE  - 16302
    ,0: POKE  - 16304,0: POKE  -
    16300,0
80  END
```

最后，运行GP 2502程序，即可看到如图2.32合并的图



图 2.32

形。应该注意的是，BLOAD命令只能把图形信息调入图形存贮区，它本身不包含显示功能，为此，用软开关设置显示方式，其中POKE-16300，0是指在高分辨第一页图形区显示图形。

图2.32也可以打印出来，只要在程序GP2502的基础上，加上几个语句即可实现，见程序GP2503。

程序GP2503清单。

```
10 D$ = CHR$ (4)
20 PRINT D$;"BLOAD GMS8"
30 PRINT D$;"BLOAD P2"
40 PRINT D$;"BRUN GP2501"
50 POKE - 16297,0: POKE - 16302
   ,0: POKE - 16304,0: POKE -
   16300,0
60 PR# 1
70 POKE 1913,1
80 PRINT CHR$ (17)
90 PR# 0
100 TEXT : HOME : END
```

其中POKE 1913,1是指第一页高分辨作图区，执行PRINT CHR$(17)↙才能打印出第一页面内的图形（或按CTRL-Q）。

如果希望既看到合并的图形，又看到另一页(指第二页)的图象，只要加分页显示程序段即可，程序GP2504可以完成这个功能。

程序GP2504清单。

```
10 D$ = CHR$ (4)
20 PRINT D$;"BLOAD GMS8"
30 PRINT D$;"BLOAD P2"
```

```
40   PRINT D$;"BRUN GP2501"
50   POKE  - 16297,0: POKE  - 16302
      ,0: POKE  - 16304,0: POKE  -
      16300,0
60   FOR I = 1 TO 1000: NEXT I
70   POKE  - 16299,0
80   FOR I = 1 TO 1000: NEXT I
90   GOTO 50
100  TEXT : HOME : END
```

如果删去GP2504中的30，40，70和80句，看到的应是GMS 8图象，见图2.33。

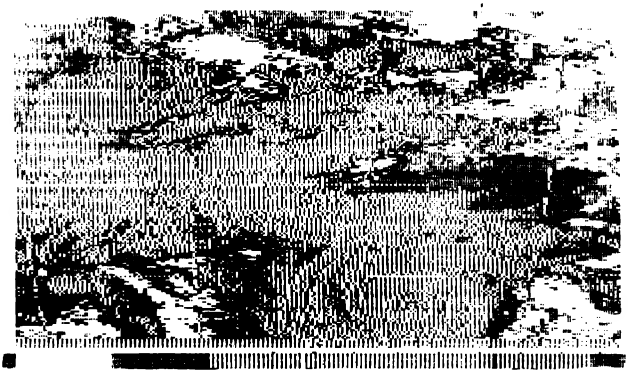


图 2.33

同理，删去GP2504中的20，40，60句，以及50句中的POKE-16300，0，再重新运行时，看到的只是一只熊猫图象。见图2.34。

应该特别指出的是，如果已经执行了GP2504程序，则内存中第一页和第二页都已存贮了图象信息，则上述两种删



图 2.34

去方法将会失效，最好的办法是关机重来，然后照删去程序段的方法处理。

以上实例说明，采用图形合并的方法，使画面更加丰富多采，而关键在于调用合并机器语言子程序GP 2501。

第二节 合并分页显示

有了上面的基本知识，我们可以重新组织程序，使画面显示方式多样，更加生动。下面提供一个程序实例，见程序GP 2505。

程序GP 2505清单

```
10 D$ =  CHR$ (4)
20  PRINT D$;"BLOAD T1"
30  PRINT D$;"BLOAD P4"
40  FOR K = 1 TO 5
50  POKE  - 16297,0: POKE  - 16302
     ,0: POKE  - 16304,0: POKE  -
     16300,0
60  FOR I = 1 TO 2000: NEXT
65  POKE  - 16299,0
70  FOR I = 1 TO 2000: NEXT
80  NEXT K
100  PRINT D$;"BRUN GP2501"
105  POKE  - 16300,0
110  FOR I = 1 TO 4000: NEXT
115 N = 300
120  POKE  - 16299,0
130  POKE  - 16300,0
132 N = N - 1
134  IF N = 0 THEN 155
140  GOTO 120
155  FOR K = 1 TO 10
160  POKE  - 16297,0: POKE  - 1630
     2,0: POKE  - 16304,0: POKE  -
     16300,0
167  FOR I = 1 TO 2000: NEXT
170  POKE  - 16299,0
175  FOR I = 1 TO 2000: NEXT
180  NEXT K
200  TEXT : HOME : PRINT ;"SEE YOU
      LETER!": END
```

运行程序GP 2505后，首先看到有两幅画分页显示，一幅是一朵鲜花，另一幅是一台计算机，它们轮流显示，每幅出现 5 次(见40—80句)。稍等片刻后，看到两幅画合在一起(见100—110句)。然后看到的是鲜花不断在计算机图案上闪烁（见115—140句)，最后看到的是分页显示合并的图案和分

开的图案(见155—180句)共10次。而当屏幕上出现“SEE YOU LETER!”时，程序结束。

下面给出运行中的几幅图案：

T 1图形见图2.35（a）

P 4 图形见图2.35（b）。

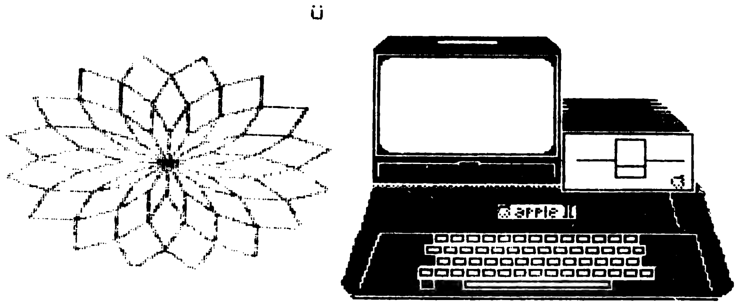


图 2.35(a)　　　　图 2.35(b)

合并图形见图2.36。

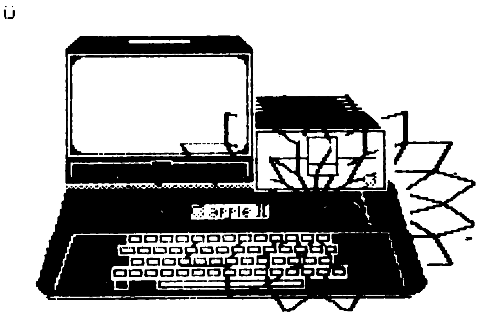


图 2.36

第三节　机器语言合并图形的方法

用机器语言编写的合并程序，其合并速度是非常快的，下面再提供一个合并程序，它存贮在＄6000—＄603A中，取名为GP 2506，并用BSAVE　GP2506，A＄6000，L＄3B存盘备用。

机器语言图形合并子程序GP 2506。

```
6000-  A9  FF  85  06  85  08  A9  1F
6008-  85  07  A9  3F  85  09  E6  06
6010-  D0  02  E6  07  E6  08  D0  02
6018-  E6  09  A2  00  A1  06  01  08
6020-  81  08  A9  3F  C5  07  D0  E6
6028-  A9  FF  C5  06  D0  E0  8D  50
6030-  C0  8D  52  C0  8D  55  C0  8D
6038-  57  C0  60
```

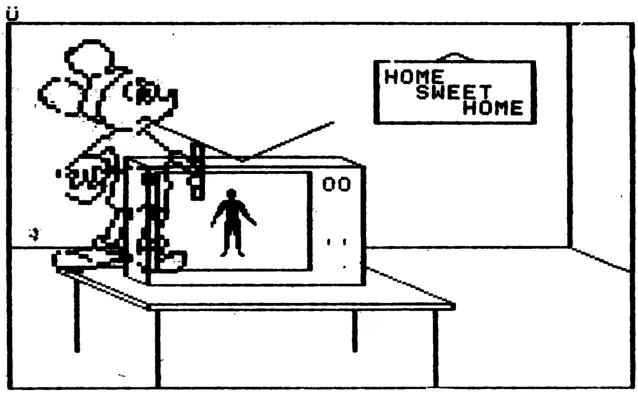


图 2.37

为了观察两幅图形的合并效果，谁按下述步骤执行：

- ·BLOAD GP2506↙
- ·BLOAD PIC1↙
- ·BLOAD MLW↙
- ·CALL 24768↙

其中图形PIC1存贮在$2000—$3FFF中，图形MLW存贮在$4000—$5FFF中。合并的图形如图2.37所示。

第四节　用BASIC 程序合并图形

图形合并也可以用BASIC语言编程实现，下面是两个实例：

实例1，是用高分辨作图语句编制的，共有5个画面，为了显示5个图形的合并过程，程序中安排了GET语句，程序运行后首先显示一个从左边开始的由上至下逐行画线的长条（见300句），待长条显示完成后，按一下任意键，在屏幕下面由左至右画隔行线（见400句），再按下任意键，屏幕右边由右到左竖直画线（见500句），再按任意键，在上方出现由左到右、由下到上隔行画线(见600句)，再次按任意键后，屏幕中央出现又一个图案(见700)，最后经710句延迟，图形结束。

这里仅仅作为一个例子，说明合并图形的一个方法，不难用此种方法，画一个辅助教学的几何图形，由于可以用手控执行程序，比较形象直观。

程序GP2507清单。

```
5   HOME
10  HGR2 : POKE 16302,0: HCOLOR= 3
```

```
20   FOR J = 0 TO 4
40   GOSUB 200
50   GET K$
70   NEXT J
200  ON J + 1 GOTO 300,400,500,600
      ,700
300  FOR I = 0 TO 190: HPLOT 0,I TO
     50,I: NEXT I: RETURN
400  FOR I = 190 TO 150 STEP  - 2:
      HPLOT 0,I TO 279,I: NEXT I
410  RETURN
500  FOR I = 279 TO 230 STEP  - 1:
      HPLOT I,0 TO I,190: NEXT I
510  RETURN
600  FOR I = 0 TO 40 STEP 2: HPLOT
     0,40 - I TO 279,40 - I: NEXT
     I: RETURN
700  FOR I = 140 TO 82 STEP  - 1
705  HPLOT I,I TO 279 - I,I TO 279
      - I,190 - I TO I,190 - I TO
     I,I: NEXT I
710  FOR I = 1 TO 3000: NEXT I: TEXT
     : HOME : END
```

运行结果，见图2.38。

实例2，编制原理同实例1，只不过开始用DOS命令调一幅已画好的图形，也是按空格键（或任意键）执行操作过程。本程序易懂，读者可以自行运行，体会其中图形合并的乐趣。

程序GP2508清单。

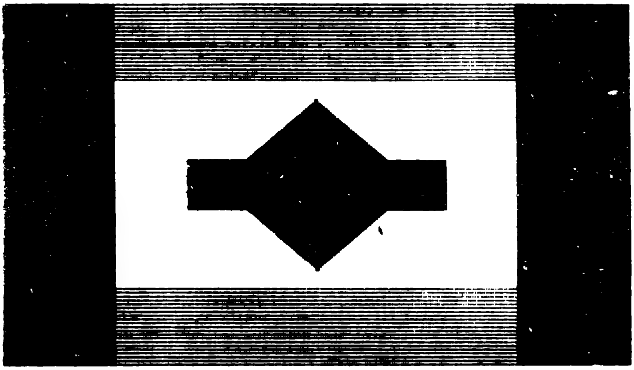


图 2.38

```
10  HOME
20  HGR : HCOLOR= 7: POKE  - 16302
     ,0
30  FOR I = 0 TO 5
40 D$ =  CHR$ (4)
50  GOSUB 90
60  GET K$
70  NEXT
80  TEXT : HOME : END
90  ON I + 1 GOTO 100,120,130,140,
     150,160
100  PRINT D$"BLOADHY.PICT"
110  GOTO 180
120  HPLOT 0,0 TO 279,0: RETURN
130  HPLOT 279,0 TO 279,190: RETURN

140  HPLOT 279,190 TO 0,190: RETURN

150  HPLOT 0,190 TO 0,0: RETURN
160  FOR I = 0 TO 40: HPLOT I,0 TO
      I,190: HPLOT 279 - I,0 TO 279
```

```
     - I,190: NEXT I
170  RETURN
180  POKE  - 16300,0: POKE  - 1630
     4,0: POKE  - 16297,0: POKE  -
     16302,0
190  GOTO 170
```

运行结果，见图2.39。

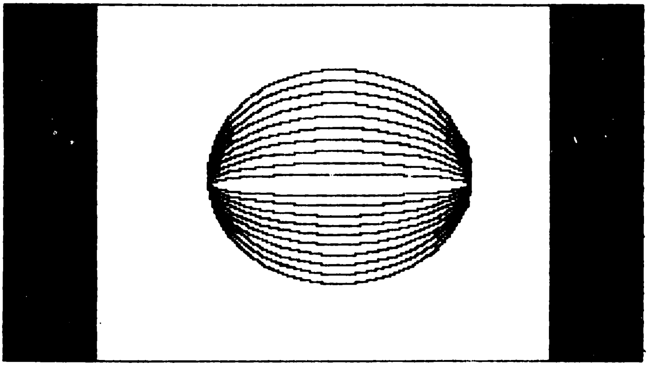


图 2.39

第五节　多幅图形的合成

利用剪辑和合并技术，可以完成多幅图形合成。例如，将4幅（原则上可以很多幅）不同的图形，经过剪辑程序处理后，再用合并技术，合成在一张画面上。

图2.40是4幅图形合成的实例。

为实现多幅图形的合成，需要两个工具软件：一个是剪辑软件(见程序DYJGCU)，它完成对单幅图形和多幅图形的剪辑、编辑加工任务；一个是合并软件(见程序HBCV)，编辑软件(见程序DYJGCU)，它完成两幅或更多幅图形的合并工作。

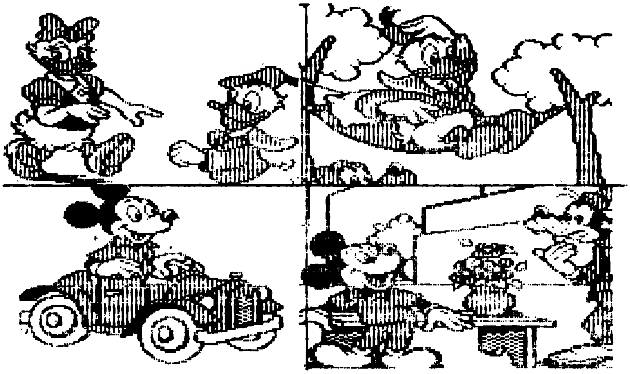


图 2.40

由于多幅图形合成技术比较复杂，现将操作步骤详述如下：

· LOAD DYJGCU↙；调剪辑编辑程序

· RUN DYJGOU↙；运行剪辑程序

· 出现INPOT PICT
NAME：按入TOAL↙；调处理的图形，如TOAL进内存

· 键入0，不用回车；对图形进行下移(按D键)、左移(按L键)、反相(按C键)、对称(按X键，或Y键）处理，直到满意为止

· 按CTRL-RESET↙；中断程序运行

· 键入CALL-151↙；进入监控状态

· ＊4000＜2000·3FFFM↙；将原图形搬至高分辨率第二页

· 按CTRL-RESET↙；返回BASIC状态
· IHGR↙；　清第一页画面
· JRUN↙
· ?　；出现？表示询问待处理图形移至何处
· 键入0,0,0,0,16，93↙；键入移动的坐标
· 按CTRL-RESET↙；中断程序执行
· BSAVE TX1, A$2000,
L$2000↙　；将剪辑后的图形存盘，文件名为TX1
· RUN 10↙，在出现
INPUT PICT NAME：
后，重复上述过程　；处理加工第二幅图形，第三幅、第四幅……等等。
· BLOAD HBCU↙　；调合并程序进内存
· BLOAD TX1, A$2000↙；调处理加工后的图形，如TX1,进第一页
· BLOAD TX2, A$4000↙；调处理加工后的图形，如TX2,进第二页
· CALL 768↙　；调合并程序
· RUN 6↙　；完成两幅图形合并
· 按CTRL-RESET↙；中断
· BSAVE TX1—TX2,
A$2000, L$2000↙　；将两幅合成的图形存盘
· BLOAD TX3, A$2000↙
· BLOAD TX4, A$2000↙

· CALL 768↙

· RUN 6↙；完成另外两幅图形，如TX3，TX 4的合并工作

· 按CTRL-RESET↙；中断

· BSAVE TX3—TX4,
A $2000, L $2000↙；将合成后的第三幅（TX3），第四幅（TX4）图形存盘

· BLOAD TX1—TX2, A $2000↙

· BLOAD TX3—TX4, A $4000↙

· CALL 768↙

· RUN 6↙　　；完成4幅图形的合成

· 按CTRL-RESET↙；中断

· BSAVE TOAL, A $2000, L $2000；存4幅合成的图形，取名为TOAL，备用

· PR#1↙　　；接通打印机

· POKE 1913, 33↙　　；选取高分辨第一页图形，即4幅合成图形

· PRINT CHR $(17)↙
或按CTRL-Q↙　　；打印4幅合成图形

上述各项操作，必须按步骤严格顺序执行，虽然比较复杂，但只要反复多练习几次，就会熟能生巧。

程序DYJGCU清单。

```
1  REM  TXJJ
2  INPUT A,B,C,D,E,F
3  POKE 0,A: POKE 1,B: POKE 6,C: POKE
     7,D: POKE 8,E: POKE 9,F
4  POKE 16,32: POKE 230,32
5  PRINT  CHR$ (4)"BRUN TXJJ1,A$60
```

```
    OO"
6  POKE  - 16297,0: POKE  - 16302,
    0: POKE  - 16304,0: POKE  - 1
    6300,0
7  END
10  REM   PRO-1-JGCU
12 D$ =  CHR$ (4)
13  PRINT D$;"BLOAD PRO"
15  PRINT D$;"BLOAD JGCU"
20  INPUT "INPUT PICT NAME:";K$
30  PRINT D$"BLOAD"K$",A$2000
50  HOME
60  POKE  - 16304,0: POKE  - 16297
    ,0: POKE  - 16300,0
70  POKE  - 16301,0: POKE 230,32
80 A$ = "ÄOÜEXIT Ä1ÜHALF PAGE NOR-
    SYMM COPY Ä2ÜHALF PAGE INV-SY
    MM COPY Ä3ÜHALF PAGE IOR-SAME
     COPY Ä4ÜHALF PAGE INV-SAME C
    OPY Ä5ÜALL PAGE NOR-SYMM COPY
     Ä6ÜALL PAGE INV-SYMM COPY Ä7
    ÜALL PAGE NOR-SAME COPY Ä8ÜAL
    L PAGE INV-SAME COPY Ä9ÜREBLO
    AD"
90  HTAB 1: VTAB 23: PRINT  LEFT$
    (A$,39);:A$ =  MID$ (A$,2) +
     LEFT$ (A$,1)
95 Q =  PEEK ( - 16384): IF Q < 12
    8 THEN  FOR Q = 1 TO 100: NEXT
    :Q =  FRE (0): GOTO 90
100  POKE  - 16368,0:Q = Q - 176: IF
    Q < 0 OR Q > 9 THEN 90
110  IF Q = 0 THEN 300
120  IF Q = 9 THEN  PRINT : GOTO 5
    0
130  FOR R = 1 TO  INT (Q / 2 + 0.
    5)
140  READ A,B,C,D,E,F,G,H,I,J,K,L,
```

```
         M,N,O,P
150   POKE 781,A: POKE 796,B: POKE
      810,C: POKE 811,D
160   POKE 812,E: POKE 813,F: POKE
      814,G: POKE 833,H
170   POKE 834,I: POKE 835,J: POKE
      836,K: POKE 851,L
180   POKE 869,M: POKE 930,N: POKE
      948,O: POKE 954,P
190   NEXT R: RESTORE
200   IF  INT (Q / 2) = Q / 2 THEN
       POKE 822,69: GOTO 220
210   POKE 822,37
220   CALL 768
230   IF Q > 4 THEN  POKE  - 16299,
      0: POKE  - 16302,0: POKE 230,
      64: FOR S = 0 TO 2000: NEXT
240   GOTO 60
250   DATA 79,119,251,162,7,70,251,
      229,235,133,254,20,64,36,32,3
      9
260   DATA  60,100,250,76,52,3,234,
      76,71,3,234,20,64,36,32,20
270   DATA 79,119,251,162,7,70,251,
      229,235,133,254,40,96,68,64,3
      9
280   DATA   40,80,250,76,52,3,234,
      76,71,3,234,40,96,68,64,0
300   CALL 24576
```

其中PRO程序清单。

```
0300- 20 A5 03 20 25 03 A9 28
0308- 85 FC 85 EE A9 4F 85 FE
0310- 85 EF 20 25 03 A9 50 85
0318- FC 85 EE A9 77 85 FE 85
```

```
0320-  EF  20  25  03  60  A0  00  B1
0328-  FC  85  FB  A2  07  46  FB  26
0330-  FA  CA  D0  F9  A9  7F  25  FA
0338-  85  FA  84  EB  38  A5  FE  85
0340-  E3  E5  EB  85  FE  A0  00  A5
0348-  FA  91  FE  A5  E3  85  FE  A4
0350-  EB  C8  C0  14  D0  D1  A5  FD
0358-  18  69  04  85  FD  A5  FF  18
0360-  69  04  85  FF  C9  40  90  BD
0368-  E6  EC  A5  EC  C9  02  F0  19
0370-  A5  FC  18  69  80  85  FC  A5
0378-  FE  18  69  80  85  FE  A5  ED
0380-  85  FD  A5  28  85  FF  4C  25
0388-  03  A0  00  84  EC  A5  EE  85
0390-  FC  A5  EF  85  FE  E6  ED  A5
0398-  ED  85  FD  E6  28  A5  28  85
03A0-  FF  C9  24  90  80  A9  00  85
03A8-  EC  85  FC  85  EE  A9  20  85
03B0-  ED  85  FD  A9  20  85  28  85
03B8-  FF  A9  27  85  FE  85  EF  60
```

程序JGCU清单。

```
6000-  AD  50  C0  AD  52  C0  AD  54
6008-  C0  AD  57  C0  A0  00  84  3A
6010-  84  3C  A0  40  84  3B  C8  84
6018-  3D  A0  BF  98  0A  0A  29  1C
6020-  85  15  98  6A  6A  6A  6A  29
6028-  03  05  15  09  20  91  3C  98
6030-  6A  29  E0  85  14  6A  6A  29
6038-  18  05  14  91  3A  88  C0  FF
6040-  D0  D9  2C  00  C0  10  FB  AD
6048-  00  C0  C9  C4  D0  3C  A0  28
6050-  88  B9  00  20  99  00  42  98
6058-  D0  F6  A2  01  BD  00  40  85
6060-  14  BD  00  41  85  15  A0  28
6068-  88  B1  14  85  3A  B9  00  42
```

```
6070- 91 14 A5 3A 99 00 42 98
6078- D0 EE 8A F0 0A E8 E0 C0
6080- D0 DA A2 00 4C 5C 60 4C
6088- 42 60 C9 CC D0 28 A2 C0
6090- CA BD 00 40 85 14 BD 00
6098- 41 85 15 A0 00 B1 14 85
60A0- 3A C8 B1 14 88 91 14 C8
60A8- C0 27 D0 F5 A5 3A 91 14
60B0- 8A D0 DD 4C 42 60 2C 10
60B8- C0 C9 D9 D0 5B A0 00 84
60C0- 11 A4 11 B9 00 40 85 14
60C8- B9 00 41 85 15 A2 27 86
60D0- 3E A0 00 A2 13 84 3A A4
60D8- 3A B1 14 85 3B 85 3C A4
60E0- 3E B1 14 85 3D 85 3F A0
60E8- 08 06 3B 6A 88 D0 FA 06
60F0- 3C 6A A4 3E 91 14 A0 08
60F8- 06 3D 6A 88 D0 FA 06 3F
6100- 6A A4 3A 91 14 C6 3E E6
6108- 3A E4 3E D0 CA E6 11 A2
6110- C0 E4 11 D0 AC 4C 42 60
6118- C9 D8 D0 40 A2 60 A0 00
6120- 84 11 A0 BF 84 12 A4 11
6128- B9 00 40 85 14 B9 00 41
6130- 85 15 A4 12 B9 00 40 85
6138- 16 B9 00 41 85 17 A0 00
6140- B1 14 85 3A B1 16 91 14
6148- A5 3A 91 16 C8 C0 28 D0
6150- EF E6 11 C6 12 E4 11 D0
6158- CD 4C 42 60 C9 C3 D0 2A
6160- A0 20 A2 40 AD 00 20 0A
6168- 49 FF 6A 8D 00 20 EE 65
6170- 61 EE 6C 61 D0 EE EE 66
6178- 61 EE 6D 61 EC 66 61 D0
6180- E3 8C 66 61 8C 6D 61 4C
6188- 42 60 C9 C5 D0 F9 60
```

合并程序HBCU清单。

```
0300- A9 00 85 00 A9 20 85 01
0308- A9 00 85 02 A9 40 85 03
0310- A0 00 B1 02 F0 02 91 00
0318- C8 D0 F7 E6 03 E6 01 A5
0320- 01 C9 40 90 ED 60
```

第六节　图形合成的简便方法

在第五节中，我们介绍了多幅图形合成的方法和详细操作步骤，不难发现上述操作过于频繁，且不易记忆。现介绍一个4幅图形合成的实例，程序和操作都十分简单。

其基本思想是，将高分辨图形第二页的画面上，剪下一块图形，贴到高分辨图形第一页的指定位置上，就可以实现多幅图形的合成。全部操作只用了两个程序：TXJJ 1和P 30-3。TXJJ 1为完成剪辑、合并的机器语言程序，P 30-3则是一个BASIC程序，它实现对剪辑、合并程序TXJJ 1的控制。

程序P 30-3可以十分方便地一次完成四幅图形的合成。该程序运行后首先清屏，然后调用200句开始的子程序，在高分辨率第1页上表演一个画线图形，接着调用机器语言子程序TXJJ1进内存，再用50—140句的循环，分别调Y1—Y4四幅图形，每调一幅图形执行一次CALL 24576，按一定坐标剪辑并放在高分辨率第1页上，当循环全部完成后，四幅图形即已合成完成。经150句延迟一段时间后，执行160句，将4幅合成的图形以文件名X存盘，最后是画线清屏，程序结束。

为方便上述操作，程序P 30-3，TXJJ 1 以及Y1— Y4 4幅图形资料，应放在同一张盘中。

程序P 30-3清单：

```
10  REM  P30-3
20 D$ =  CHR$ (4): HGR : POKE  - 1
     6302,0
30  PRINT D$"BLOAD TXJJ1"
40  HCOLOR= 7: FOR I = 0 TO 95: HPLOT
     0,I TO 141,I: HPLOT 141,191 -
     I TO 279,191 - I: NEXT I
45  FOR I = 0 TO 95: HPLOT 0,191 -
     I TO 141,191 - I: HPLOT 141,I
      TO 279,I: NEXT I
50  FOR I = 1 TO 4
60  PRINT D$"BLOAD G";I;",A$4000"
70  IF I = 1 THEN A = 0:B = 0:C =
     0:D = 0:E = 19:F = 95
72  IF I = 2 THEN A = 19:B = 0:C =
     19:D = 0:E = 39:F = 95
74  IF I = 3 THEN A = 0:B = 95:C =
     0:D = 95:E = 19:F = 191
76  IF I = 4 THEN A = 19:B = 95:C =
     19:D = 95:E = 39:F = 191
78  POKE 0,A: POKE 1,B: POKE 6,C: POKE
     7,D: POKE 8,E: POKE 9,F: POKE
     16,32: POKE 230,32: CALL 2457
     6
80  NEXT I
100  FOR I = 1 TO 2000: NEXT I
120  TEXT : HOME : END
```

变量说明：A、B 为要搬移到目的区域的左上角坐标，前者为X方向，后者为Y方向(下同)，它们存放在＄00和＄01

单元中，在BASIC程序中用POKE 0，A:POKE 1,B设置。

C、D为要搬移区域的左上角坐标，它们分别存放在$06和$07单元中，同理用POKE 6，C:POKE 7,D设置。

E、F为要搬移区域的右下角坐标，分别存放在$08和$09单元中，用POKE 8,E:POKE 9，F设置。

而在$10单元中，存放的是页面值，当值为$00时，在高分辨第一页图形面内搬移；当值为$20时，则从高分辨第二页图形，剪贴到第一页。

$E6单元中存放的值，至关重要，第一页图形放$20，第二页图形放$40。

若要改放剪贴图形的位置，可改动A、B、C、D、E、F值，其坐标对应屏幕位置如下：

($0，$0)：对应屏幕左上角

($0，$C0)：对应屏幕左下角

($27，$0)：对应屏幕右上角

($27，$C0)：对应屏幕右下角

($13，$60)：对应屏幕中心位置

改变A—F坐标值，不难在一幅屏幕上合成8幅以及更多幅图形来。

机器语言程序TXJJ1清单：

```
8000- A0 00 98 48 20 11 F4 68
8008- A8 A5 26 99 00 8E A5 27
8010- 99 00 8F C8 C0 C0 D0 EA
8018- A4 07 B9 00 8E 85 0A B9
8020- 00 8F 18 65 10 85 0B A4
```

```
8028- 01 B9 00 8E 85 0C B9 00
8030- 8F 85 0D A5 06 85 0E A5
8038- 00 85 0F A4 0E B1 0A A4
8040- 0F 91 0C E6 0F A5 0F C9
8048- 28 F0 08 E6 0E A5 0E C5
8050- 08 D0 E8 E6 01 A5 01 C9
8058- C0 F0 08 E6 07 A5 07 C5
8060- 09 D0 B5 60
```

第六章　移动显示

在使用高分辨第一页及第二页绘制图形时，用一般的显示方法比较单调和死板。如果我们编制一些机器语言子程序，在需要时调用它们，就会使图象显示多样化，其效果也就好得多。事实上，许多游戏软件常采用多样化的显示技巧，其中用得最多的就是让图象在移动过程中显示，给人以明快深刻的印象。

本章主要介绍上、下移动，左、右移动显示的机器语言子程序，并详细剖析其原理，供读者在此基础上应用和改进。这些子程序也可以方便地移植，从而使您的软件增辉添色。

第一节　画面结构与其记忆位置的对应关系

在介绍各种机器语言子程序之前，我们先谈谈中华学习机画面结构与其记忆位置的对应关系。同APPLEI机一样，其画面结构比较特殊，显示器地址分配如图2.41所示。

由图2.41可知，屏幕上的画面结构与其记忆位置不是顺序一一对应的关系。但是我们可用下面的公式来计算其对应的记忆地址：

AD＝8192＋A＊1024＋B＊128＋C＊40＋Y

式中：A＝X－8＊D

B＝D－8＊C

C＝INT(D/8)

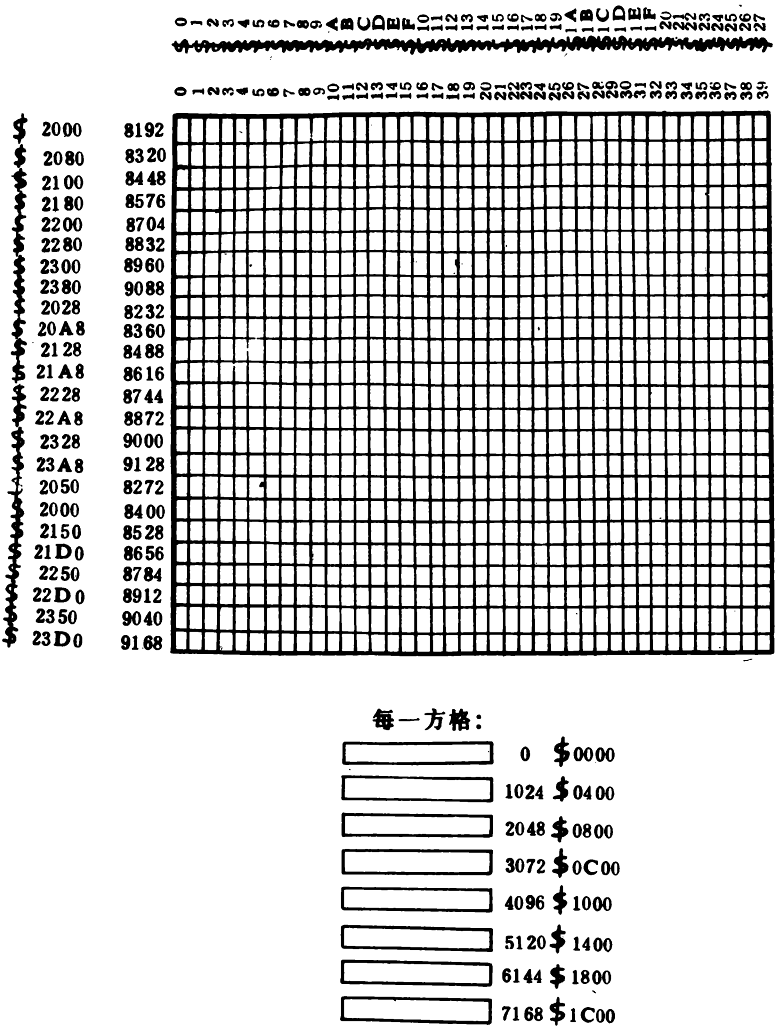

$2000 8192
$2080 8320
$2100 8448
$2180 8576
$2200 8704
$2280 8832
$2300 8960
$2380 9088
$2028 8232
$20A8 8360
$2128 8488
$21A8 8616
$2228 8744
$22A8 8872
$2328 9000
$23A8 9128
$2050 8272
$2000 8400
$2150 8528
$21D0 8656
$2250 8784
$22D0 8912
$2350 9040
$23D0 9168
每一方格:
0 $0000
1024 $0400
2048 $0800
3072 $0C00
4096 $1000
5120 $1400
6144 $1800
7168 $1C00

图 2.41

$$D=INT(X/8)$$

公式中某一显示点的坐标为（X，Y），由行坐标X，列坐标Y可直接计算该坐标所对应的记忆地址AD。我们可用BASIC语言编程（也可用机器语言编程）算好地址与程序存放在一起，然后采用查表的方法读取地址以提高运行速度。程序GP 2601是用BASIC语言编程，程序GP 2602是用机器语言编程，它们都可以将第一页每一行的首地址计算好并存放在$8100（存放高八位）和$8200（存放低八位）以后的各单元中。

```
10  FOR X = 0 TO 191
20 D =  INT (X / 8)
30 C =  INT (D / 8)
40 B = D - 8 * C
50 A = X - 8 * D
60 AD = 8192 + A * 1024 + B * 128 +
    C * 40
70  POKE 33024 + X, INT (AD / 256)

80  POKE 33280 + X,AD -  INT (AD /
    256) * 256
90  NEXT
100  END
```

运行程序GP 2601后，在监控状态下可以看到第一页每一行的首地址存贮情况。

```
8100- 20 24 28 2C 30 34 38 3C
8108- 20 24 28 2C 30 34 38 3C
8110- 21 25 29 2D 31 35 39 3D
8118- 21 25 29 2D 31 35 39 3D
8120- 22 26 2A 2E 32 36 3A 3E
8128- 22 26 2A 2E 32 36 3A 3E
```

8130-	23	27	2B	2F	33	37	3B	3F
8138-	23	27	2B	2F	33	37	3B	3F
8140-	20	24	28	2C	30	34	38	3C
8148-	20	24	28	2C	30	34	38	3C
8150-	21	25	29	2D	31	35	39	3D
8158-	21	25	29	2D	31	35	39	3D
8160-	22	26	2A	2E	32	36	3A	3E
8168-	22	26	2A	2E	32	36	3A	3E
8170-	23	27	2B	2F	33	37	3B	3F
8178-	23	27	2B	2F	33	37	3B	3F
8180-	20	24	28	2C	30	34	38	3C
8188-	20	24	28	2C	30	34	38	3C
8190-	21	25	29	2D	31	35	39	3D
8198-	21	25	29	2D	31	35	39	3D
81A0-	22	26	2A	2E	32	36	3A	3E
81A8-	22	26	2A	2E	32	36	3A	3E
81B0-	23	27	2B	2F	33	37	3B	3F
81B8-	23	27	2B	2F	33	37	3B	3F
8200-	00	00	00	00	00	00	00	00
8208-	80	80	80	80	80	80	80	80
8210-	00	00	00	00	00	00	00	00
8218-	80	80	80	80	80	80	80	80
8220-	00	00	00	00	00	00	00	00
8228-	80	80	80	80	80	80	80	80
8230-	00	00	00	00	00	00	00	00
8238-	80	80	80	80	80	80	80	80
8240-	28	28	28	28	28	28	28	28
8248-	A8	A8	A8	A8	A8	A8	A8	A8
8250-	28	28	28	28	28	28	28	28
8258-	A8	A8	A8	A8	A8	A8	A8	A8
8260-	28	28	28	28	28	28	28	28
8268-	A8	A8	A8	A8	A8	A8	A8	A8
8270-	28	28	28	28	28	28	28	28
8278-	A8	A8	A8	A8	A8	A8	A8	A8
8280-	50	50	50	50	50	50	50	50
8288-	D0	D0	D0	D0	D0	D0	D0	D0

```
8290-  50 50 50 50 50 50 50 50
8298-  D0 D0 D0 D0 D0 D0 D0 D0
82A0-  50 50 50 50 50 50 50 50
82A8-  D0 D0 D0 D0 D0 D0 D0 D0
82B0-  50 50 50 50 50 50 50 50
82B8-  D0 D0 D0 D0 D0 D0 D0 D0
```

在监控状态下，健入机器语言子程序GP 2602。

```
*8000LL

8000-    A9 00          LDA   #$00
8002-    85 06          STA   $06
8004-    85 08          STA   $08
8006-    8D 19 80       STA   $8019
8009-    8D 22 80       STA   $8022
800C-    A9 82          LDA   #$82
800E-    85 07          STA   $07
8010-    A9 81          LDA   #$81
8012-    85 09          STA   $09
8014-    A9 20          LDA   #$20
8016-    85 E6          STA   $E6
8018-    A9 C0          LDA   #$C0
801A-    A0 00          LDY   #$00
801C-    A2 00          LDX   #$00
801E-    20 11 F4       JSR   $F411
8021-    A0 C0          LDY   #$C0
8023-    A5 26          LDA   $26
8025-    91 06          STA   ($06),Y
8027-    A5 27          LDA   $27
8029-    91 08          STA   ($08),Y
802B-    EE 19 80       INC   $8019
802E-    EE 22 80       INC   $8022
8031-    C0 BF          CPY   #$BF
8033-    90 E3          BCC   $8018
8035-    60             RTS
```

然后我们从＄8100—＄81BF及＄8200—＄82BF中看到和运行GP2601后一致的结果。

```
8100- 20 24 28 2C 30 34 38 3C
8108- 20 24 28 2C 30 34 38 3C
8110- 21 25 29 2D 31 35 39 3D
8118- 21 25 29 2D 31 35 39 3D
8120- 22 26 2A 2E 32 36 3A 3E
8128- 22 26 2A 2E 32 36 3A 3E
8130- 23 27 2B 2F 33 37 3B 3F
8138- 23 27 2B 2F 33 37 3B 3F
8140- 20 24 28 2C 30 34 38 3C
8148- 20 24 28 2C 30 34 38 3C
8150- 21 25 29 2D 31 35 39 3D
8158- 21 25 29 2D 31 35 39 3D
8160- 22 26 2A 2E 32 36 3A 3E
8168- 22 26 2A 2E 32 36 3A 3E
8170- 23 27 2B 2F 33 37 3B 3F
8178- 23 27 2B 2F 33 37 3B 3F
8180- 20 24 28 2C 30 34 38 3C
8188- 20 24 28 2C 30 34 38 3C
8190- 21 25 29 2D 31 35 39 3D
8198- 21 25 29 2D 31 35 39 3D
81A0- 22 26 2A 2E 32 36 3A 3E
81A8- 22 26 2A 2E 32 36 3A 3E
81B0- 23 27 2B 2F 33 37 3B 3F
81B8- 23 27 2B 2F 33 37 3B 3F
8200- 00 00 00 00 00 00 00 00
8208- 80 80 80 80 80 80 80 80
8210- 00 00 00 00 00 00 00 00
8218- 80 80 80 80 80 80 80 80
8220- 00 00 00 00 00 00 00 00
8228- 80 80 80 80 80 80 80 80
8230- 00 00 00 00 00 00 00 00
```

```
8238- 80 80 80 80 80 80 80 80
8240- 28 28 28 28 28 28 28 28
8248- A8 A8 A8 A8 A8 A8 A8 A8
8250- 28 28 28 28 28 28 28 28
8258- A8 A8 A8 A8 A8 A8 A8 A8
8260- 28 28 28 28 28 28 28 28
8268- A8 A8 A8 A8 A8 A8 A8 A8
8270- 28 28 28 28 28 28 28 28
8278- A8 A8 A8 A8 A8 A8 A8 A8
8280- 50 50 50 50 50 50 50 50
8288- D0 D0 D0 D0 D0 D0 D0 D0
8290- 50 50 50 50 50 50 50 50
8298- D0 D0 D0 D0 D0 D0 D0 D0
82A0- 50 50 50 50 50 50 50 50
82A8- D0 D0 D0 D0 D0 D0 D0 D0
82B0- 50 50 50 50 50 50 50 50
82B8- D0 D0 D0 D0 D0 D0 D0 D0
```

程序GP 2602的原理说明如下：

\$8000—\$8013	初始化。设置地址存放单位，即将\$8200存放于\$07和\$06单元中，将\$8100存放于\$09和\$08单元中。设置计数单元\$8019和\$8022，用于判断X行是否算完。
\$8014—\$8017	将\$20放入\$E6单元，指定计算第一页图形显示地址。
\$8018—\$8020	调用BASIC解释程序中的子程序计算各行首地址，算好后放入\$26(低八位)和\$27(高八位)单元中。
\$8021—\$802A	将算好的各行首地址放入\$8200和\$8100以后的各单元中。

$802B—$8034	计数器加1，判断各行是否全部算完，未算完转$8018继续算。
$8035	算完结束返回。

第二节　由下向上移动显示

要实现画面由下向上移动显示，可以采用逐行向上搬移的方法，原理见图2.42。图中每一方块的含义如图2.41所示。

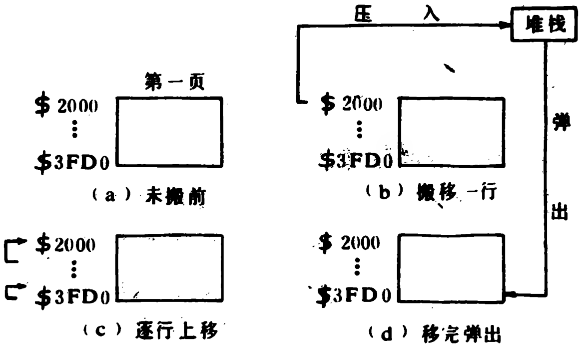


图 2.42

程序GP2603清单。

```
8000-     A2 00          LDX    #$00
8002-     BD 00 81       LDA    $8100,X
8005-     85 07          STA    $07
8007-     BD 00 82       LDA    $8200,X
800A-     85 06          STA    $06
800C-     E8             INX
800D-     BD 00 81       LDA    $8100,X
```

```
8010-    85 09       STA   $09
8012-    BD 00 82    LDA   $8200,X
8015-    85 08       STA   $08
8017-    E0 01       CPX   #$01
8019-    D0 08       BNE   $8023
801B-    A0 27       LDY   #$27
801D-    B1 06       LDA   ($06),Y
801F-    48          PHA
8020-    88          DEY
8021-    10 FA       BPL   $801D
8023-    A0 27       LDY   #$27
8025-    B1 08       LDA   ($08),Y
8027-    91 06       STA   ($06),Y
8029-    88          DEY
802A-    10 F9       BPL   $8025
802C-    E0 C0       CPX   #$C0
802E-    D0 D2       BNE   $8002
8030-    A0 00       LDY   #$00
8032-    68          PLA
8033-    91 06       STA   ($06),Y
8035-    C8          INY
8036-    C0 28       CPY   #$28
8038-    90 F8       BCC   $8032
803A-    60          RTS
803B-    A9 20       LDA   #$20
803D-    85 E6       STA   $E6
803F-    A2 00       LDX   #$00
8041-    8A          TXA
8042-    48          PHA
8043-    20 11 F4    JSR   $F411
8046-    68          PLA
8047-    AA          TAX
8048-    A5 26       LDA   $26
804A-    9D 00 82    STA   $8200,X
804D-    A5 27       LDA   $27
804F-    9D 00 81    STA   $8100,X
```

```
8052-   E8          INX
8053-   E0 C0       CPX   #$C0
8055-   D0 EA       BNE   $8041
8057-   AD 54 C0    LDA   $C054
805A-   AD 50 C0    LDA   $C050
805D-   AD 57 C0    LDA   $C057
8060-   AD 52 C0    LDA   $C052
8063-   AD 10 C0    LDA   $C010
8066-   20 00 80    JSR   $8000
8069-   AD 00 C0    LDA   $C000
806C-   10 F5       BPL   $8063
806E-   60          RTS
```

程序GP 2603在零页设置了4个单元，各单元作用分别为：

$07，$06——存放移动后目的行地址，$07存放高八位，$06存放低八位。

$09，$08——存放被移动行地址，$09存放高八位，$08存放低八位。

程序原理说明如下：

$8000—$800B	查表，取得移动后目的行地址，高八位放入$07单元，低八位放入$06单元，开始即为首地址$2000。
$800C—$8016	查表指针加1后，取得被移动行地址，高八位放入$09单元，低八位放入$08单元，开始即为$2400。
$8017—$8022	判断是否为首行，如是则将首行逐列压入堆栈，如图2.42(b)，如不是则跳过此段转$8023往下执行。开

	始即为＄2000—＄2027单元的内容压入栈内。
＄8023—＄802B	将一行逐列移动，如图2.42(c)，开始即为＄2400—＄2427单元的内容搬移至＄2000—＄2027单元中。
＄802C—＄802F	判断整幅画面是否全部上移了一行，未移完则转至 ＄8002 继续往下移，如图2.42(c)。
＄8030—＄8039	整幅画面移完后，则从堆栈中将首行内容逐列弹出至最后一行，如图2.42(d)。
＄803A	结束返回。

至此，连续不断地运行GP2603，画面即形成不断由下向上移动显示。

GP2603中＄803B—＄806E一段子程序是为演示GP2603程序而编制的，其中＄803B—＄8056与GP2602程序基本相同，是用来计算各行记忆地址的；＄8057—＄806E是在第一页用软开关设置画面显示，并控制程序的中断运行。

演示GP2603程序，可用GP2604程序。当画面不断移动时，按任意键中断返回。图2.43为中断后的画面。

程序GP2604清单。

```
10 D$ =  CHR$ (4)
20  PRINT D$;"BLOAD PIC-1,A$2000"
30  PRINT D$;"BLOAD GP2603"
40  CALL 32827
50  FOR I = 1 TO 1000: NEXT I
60  TEXT : END
```

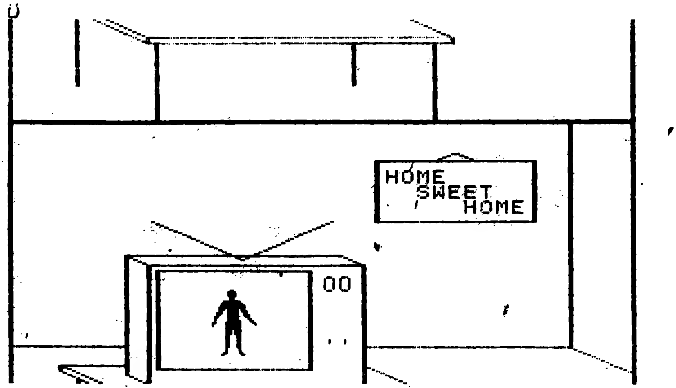


图 2.43

GP2603子程序介绍的是在第一页范围内移动，我们还可以将图形存放在第二页，逐行搬至第一页实现向上移动显示。GP2605子程序即有此功能。

```
8000-   A9 00       LDA  #$00
8002-   85 1F       STA  $1F
8004-   A2 00       LDX  #$00
8006-   BD 00 81    LDA  $8100,X
8009-   85 07       STA  $07
800B-   BD 00 82    LDA  $8200,X
800E-   85 06       STA  $06
8010-   E8          INX
8011-   BD 00 81    LDA  $8100,X
8014-   85 09       STA  $09
8016-   BD 00 82    LDA  $8200,X
8019-   85 08       STA  $08
801B-   20 45 80    JSR  $8045
801E-   E0 BF       CPX  #$BF
8020-   D0 E4       BNE  $8006
```

```
8022-   A5 08       LDA   $08
8024-   85 06       STA   $06
8026-   A5 09       LDA   $09
8028-   85 07       STA   $07
802A-   A6 1F       LDX   $1F
802C-   BD 00 81    LDA   $8100,X
802F-   18          CLC
8030-   69 20       ADC   #$20
8032-   85 09       STA   $09
8034-   BD 00 82    LDA   $8200,X
8037-   85 08       STA   $08
8039-   20 45 80    JSR   $8045
803C-   E6 1F       INC   $1F
803E-   A5 1F       LDA   $1F
8040-   C9 C0       CMP   #$C0
8042-   D0 C0       BNE   $8004
8044-   60          RTS
8045-   A0 00       LDY   #$00
8047-   B1 08       LDA   ($08),Y
8049-   91 06       STA   ($06),Y
804B-   C8          INY
804C-   C0 28       CPY   #$28
804E-   D0 F7       BNE   $8047
8050-   60          RTS
8051-   A9 20       LDA   #$20
8053-   85 E6       STA   $E6
8055-   A2 00       LDX   #$00
8057-   8A          TXA
8058-   48          PHA
8059-   20 11 F4    JSR   $F411
805C-   68          PLA
805D-   AA          TAX
805E-   A5 26       LDA   $26
8060-   9D 00 82    STA   $8200,X
8063-   A5 27       LDA   $27
8065-   9D 00 81    STA   $8100,X
8068-   E8          INX
```

```
8069-   E0 C0       CPX   #$C0
806B-   D0 EA       BNE   $8057
806D-   AD 54 C0    LDA   $C054
8070-   AD 50 C0    LDA   $C050
8073-   AD 57 C0    LDA   $C057
8076-   AD 52 C0    LDA   $C052
8079-   AD 10 C0    LDA   $C010
807C-   20 00 80    JSR   $8000
807F-   AD 00 C0    LDA   $C000
8082-   10 F5       BPL   $8079
8084-   60          RTS
```

GP 2605子程序原理见图2.44。

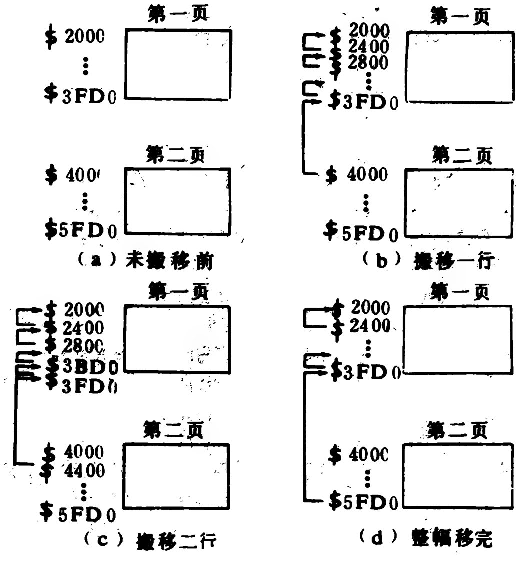


图 2.44

GP2605子程序在零页设置了5个单元，除$06—$09 4个单元的作用同GP2603子程序外，还设置了$1F单元作计数器用，以判断整幅图形移动完了没有。

程序GP2605原理说明如下：

$8000—$8003	设置计数器初始值0。
$8004—$801A	同GP2603子程序$8000—$8016。
$801B—$801D	调用逐列移动一行子程序。
$801E—$8021	判断第一页是否全部上移了一行，未移完则转至$8006继续移。
$8022—$8029	移完，将第一页末行地址作为目的行地址。
$802A—$8038	查表，将高八位加$20即取第二页被移行地址。
$8039—$803B	调用逐列移动一行子程序，如图2.44(b)。
$803C—$8043	计数器加1，判断是否全部移完，未完转$8004继续移第二行，如图2.44(c)，循环直至全部移完如图2.44(d)。
$8044	移完返回。
$8045—$8050	逐列移动一行子程序，即将被移行各列的内容逐列移至目的行各列。
$8051—$8084	同GP2603子程序$803B—$806E，演示用。

演示GP2605子程序，可用程序GP2606。

```
10 D$ =  CHR$ (4)
20  PRINT D$;"BLOAD PIC-1,A$4000"
30  PRINT D$;"BLOAD GP2605"
40  CALL 32849
50  FOR I = 1 TO 1000: NEXT I
60  TEXT : END
```

第三节　由上向下移动显示

掌握了由下向上移动显示的原理后，要实现由上向下移动显示就很方便了。最简单的方法就是不改动子程序，而将图形每行首地址颠倒放置，即改动一下计算地址的程序，见GP2607。

GP2607，$8000—$803A为由下向上移动子程序GP2603的子程序，$803B—$8063改动了一下，它计算出的图形每行首地址颠倒放置了，运行程序GP2607清单。

```
8000-    A2 00       LDX    #$00
8002-    BD 00 81    LDA    $8100,X
8005-    85 07       STA    $07
8007-    BD 00 82    LDA    $8200,X
800A-    85 06       STA    $06
800C-    E8          INX
800D-    BD 00 81    LDA    $8100,X
8010-    85 09       STA    $09
8012-    BD 00 82    LDA    $8200,X
8015-    85 08       STA    $08
8017-    E0 01       CPX    #$01
8019-    D0 08       BNE    $8023
801B-    A0 27       LDY    #$27
801D-    B1 06       LDA    ($06),Y
801F-    48          PHA
8020-    88          DEY
```

```
8021-   10 FA       BPL   $801D
8023-   A0 27       LDY   #$27
8025-   B1 08       LDA   ($08),Y
8027-   91 06       STA   ($06),Y
8029-   88          DEY
802A-   10 F9       BPL   $8025
802C-   E0 C0       CPX   #$C0
802E-   D0 D2       BNE   $8002
8030-   A0 00       LDY   #$00
8032-   68          PLA
8033-   91 06       STA   ($06),Y
8035-   C8          INY
8036-   C0 28       CPY   #$28
8038-   90 F8       BCC   $8032
803A-   60          RTS
803B-   A9 00       LDA   #$00
803D-   8D 4A 80    STA   $804A
8040-   A9 BF       LDA   #$BF
8042-   8D 4F 80    STA   $804F
8045-   A9 20       LDA   #$20
8047-   85 E6       STA   $E6
8049-   A9 00       LDA   #$00
804B-   20 11 F4    JSR   $F411
804E-   A2 BF       LDX   #$BF
8050-   A5 26       LDA   $26
8052-   9D 00 82    STA   $8200,X
8055-   A5 27       LDA   $27
8057-   9D 00 81    STA   $8100,X
805A-   EE 4A 80    INC   $804A
805D-   CE 4F 80    DEC   $804F
8060-   E0 00       CPX   #$00
8062-   D0 E5       BNE   $8049
8064-   AD 50 C0    LDA   $C050
8067-   AD 57 C0    LDA   $C057
806A-   AD 52 C0    LDA   $C052
806D-   AD 10 C0    LDA   $C010
8070-   20 00 80    JSR   $8000
```

```
8073-    AD 00 C0       LDA    $C000
8076-    10 F5          BPL    $806D
8078-    60             RTS
```

GP 2607，可以看到，图形由上向下移动显示了。

被改动部分的原理说明如下：

$ 803B —$ 8044	初始化，让其从$ 0开始计算，从$ BF开始存放。
$ 8045—$ 8048	指定计算第一页图形显示地址。
$ 8049—$ 804D	调用BASIC解释程序中的子程序计算各行首地址，算好后放入$ 26和$ 27单元中。
$ 804E —$ 8059	将算好的各行首地址颠倒放置在$ 8200和$ 8100以后的各单元中。
$ 805A—$ 8063	$ 804A加1（算下一个地址），$ 804F减一（倒着放下一个地址），判断是否全部算完，未算完转$ 8049继续算，算完往下执行演示程序。

演示GP 2607程序，将GP 2604程序中调GP 2603文件改为调GP 2607文件即可。

除了上述方法，还可以改动GP 2603子程序来实现由上向下移动显示图形，见程序GP2608。

对照GP 2608与GP 2603，可以看到$ 8001单元内容$ 00改为$ BF，$ 800C单元内容改为$ CA，$ 8018单元内容改为$ BE，$ 802D单元改为$ FF。程序GP 2608将图形最末行先压入堆栈，然后一行一行向下搬移，移完后从堆栈

弹出到图形首行，循环搬移就实现了由上向下移动显示图形。

同理，将GP2605子程序改为GP2609子程序，即可实现将第二页图形搬至第一页由上向下移动显示。

演示GP2608GP2609可分别将GP2604与GP2606中调GP2603，GP2605文件改为调GP2608、GP2609文件。

程序GP2608、GP2609清单。

```
8000-   A2 BF       LDX   #$BF
8002-   BD 00 81    LDA   $8100,X
8005-   85 07       STA   $07
8007-   BD 00 82    LDA   $8200,X
800A-   85 06       STA   $06
800C-   CA          DEX
800D-   BD 00 81    LDA   $8100,X
8010-   85 09       STA   $09
8012-   BD 00 82    LDA   $8200,X
8015-   85 08       STA   $08
8017-   E0 BE       CPX   #$BE
8019-   D0 08       BNE   $8023
801B-   A0 27       LDY   #$27
801D-   B1 06       LDA   ($06),Y
801F-   48          PHA
8020-   88          DEY
8021-   10 FA       BPL   $801D
8023-   A0 27       LDY   #$27
8025-   B1 08       LDA   ($08),Y
8027-   91 06       STA   ($06),Y
8029-   88          DEY
802A-   10 F9       BPL   $8025
802C-   E0 FF       CPX   #$FF
802E-   D0 D2       BNE   $8002
```

```
8030-   A0 00       LDY   #$00
8032-   68          PLA
8033-   91 06       STA   ($06),Y
8035-   C8          INY
8036-   C0 28       CPY   #$28
8038-   90 F8       BCC   $8032
803A-   60          RTS
803B-   A9 20       LDA   #$20
803D-   85 E6       STA   $E6
803F-   A2 00       LDX   #$00
8041-   8A          TXA
8042-   48          PHA
8043-   20 11 F4    JSR   $F411
8046-   68          PLA
8047-   AA          TAX
8048-   A5 26       LDA   $26
804A-   9D 00 82    STA   $8200,X
804D-   A5 27       LDA   $27
804F-   9D 00 81    STA   $8100,X
8052-   E8          INX
8053-   E0 C0       CPX   #$C0
8055-   D0 EA       BNE   $8041
8057-   AD 54 C0    LDA   $C054
805A-   AD 50 C0    LDA   $C050
805D-   AD 57 C0    LDA   $C057
8060-   AD 52 C0    LDA   $C052
8063-   AD 10 C0    LDA   $C010
8066-   20 00 80    JSR   $8000
8069-   AD 00 C0    LDA   $C000
806C-   10 F5       BPL   $8063
806E-   60          RTS
8000-   A9 BF       LDA   #$BF
8002-   85 1F       STA   $1F
8004-   A2 BF       LDX   #$BF
8006-   BD 00 81    LDA   $8100,X
8009-   85 07       STA   $07
```

```
800B-   BD 00 82    LDA   $8200,X
800E-   85 06       STA   $06
8010-   CA          DEX
8011-   BD 00 81    LDA   $8100,X
8014-   85 09       STA   $09
8016-   BD 00 82    LDA   $8200,X
8019-   85 08       STA   $08
801B-   20 45 80    JSR   $8045
801E-   E0 00       CPX   #$00
8020-   D0 E4       BNE   $8006
8022-   A5 08       LDA   $08
8024-   85 06       STA   $06
8026-   A5 09       LDA   $09
8028-   85 07       STA   $07
802A-   A6 1F       LDX   $1F
802C-   BD 00 81    LDA   $8100,X
802F-   18          CLC
8030-   69 20       ADC   #$20
8032-   85 09       STA   $09
8034-   BD 00 82    LDA   $8200,X
8037-   85 08       STA   $08
8039-   20 45 80    JSR   $8045
803C-   C6 1F       DEC   $1F
803E-   A5 1F       LDA   $1F
8040-   C9 FF       CMP   #$FF
8042-   D0 C0       BNE   $8004
8044-   60          RTS
8045-   A0 00       LDY   #$00
8047-   B1 08       LDA   ($08),Y
8049-   91 06       STA   ($06),Y
804B-   C8          INY
804C-   C0 28       CPY   #$28
804E-   D0 F7       BNE   $8047
8050-   60          RTS
8051-   A9 20       LDA   #$20
8053-   85 E6       STA   $E6
8055-   A2 00       LDX   #$00
```

```
8057-   8A          TXA
8058-   48          PHA
8059-   20 11 F4    JSR   $F411
805C-   68          PLA
805D-   AA          TAX
805E-   A5 26       LDA   $26
8060-   9D 00 82    STA   $8200,X
8063-   A5 27       LDA   $27
8065-   9D 00 81    STA   $8100,X
8068-   E8          INX
8069-   E0 C0       CPX   #$C0
806B-   D0 EA       BNE   $8057
806D-   AD 54 C0    LDA   $C054
8070-   AD 50 C0    LDA   $C050
8073-   AD 57 C0    LDA   $C057
8076-   AD 52 C0    LDA   $C052
8079-   AD 10 C0    LDA   $C010
807C-   20 00 80    JSR   $8000
807F-   AD 00 C0    LDA   $C000
8082-   10 F5       BPL   $8079
8084-   60          RTS
```

第四节 向左、向右移动显示

以上几节介绍的都是上下移动显示子程序，本节介绍左右移动显示子程序。

GP2610为向左移动显示子程序，其原理见图2.45。

GP2610子程序采用逐列向左搬移的方法实现图象向左移动显示，其程序原理说明如下：

$8000—$800D　　查表取各行首地址。

$800E—$8015　　各行0列压栈见图2.45（b)。

$8016—$8021　　再取各行首地址。

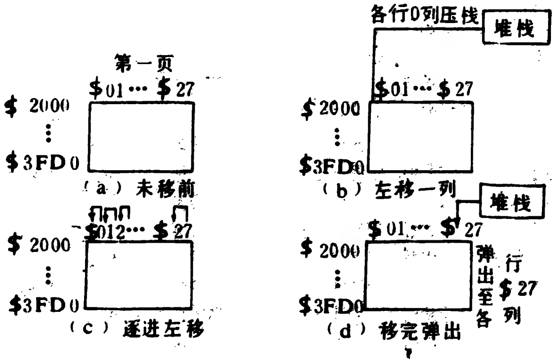


图 2.45

$8022—$8025	判各列是否全部移完，移完转$8036，未完继续。
$8026—$8030	逐列左移见图2.45（c）。
$8031—$8035	列＋1，兼做计数用，转$8016循环。
$8036—$803D	当$27列移完后，从堆栈弹出至各行$27，见图2.45（d）。
$803E	结束返回。
$803F—$8072	演示用。

重复调用本程序，图象显示即不断向左移动。运行中的一个画面见图2.46。

要使图象向右移动显示，则只要先将各行$27列内容压栈，然后逐列右移，移完再从堆栈程序GP2610清单。

```
8000-    A0 00       LDY   #$00
8002-    A2 BF       LDX   #$BF
8004-    BD 00 81    LDA   $8100,X
```

```
8007-    85 07       STA   $07
8009-    BD 00 82    LDA   $8200,X
800C-    85 06       STA   $06
800E-    B1 06       LDA   ($06),Y
8010-    48          PHA
8011-    CA          DEX
8012-    E0 FF       CPX   #$FF
8014-    D0 EE       BNE   $8004
8016-    A2 00       LDX   #$00
8018-    BD 00 81    LDA   $8100,X
801B-    85 07       STA   $07
801D-    BD 00 82    LDA   $8200,X
8020-    85 06       STA   $06
8022-    C0 27       CPY   #$27
8024-    F0 10       BEQ   $8036
8026-    C8          INY
8027-    B1 06       LDA   ($06),Y
8029-    88          DEY
802A-    91 06       STA   ($06),Y
802C-    E8          INX
802D-    E0 C0       CPX   #$C0
802F-    D0 E7       BNE   $8018
8031-    C8          INY
8032-    C0 28       CPY   #$28
8034-    D0 E0       BNE   $8016
8036-    68          PLA
8037-    91 06       STA   ($06),Y
8039-    E8          INX
803A-    E0 C0       CPX   #$C0
803C-    90 DA       BCC   $8018
803E-    60          RTS
803F-    A9 20       LDA   #$20
8041-    85 E6       STA   $E6
8043-    A2 00       LDX   #$00
8045-    8A          TXA
8046-    48          PHA
8047-    20 11 F4    JSR   $F411
```

```
804A-   68          PLA
804B-   AA          TAX
804C-   A5 26       LDA   $26
804E-   9D 00 82    STA   $8200,X
8051-   A5 27       LDA   $27
8053-   9D 00 81    STA   $8100,X
8056-   E8          INX
8057-   E0 C0       CPX   #$C0
8059-   D0 EA       BNE   $8045
805B-   AD 54 C0    LDA   $C054
805E-   AD 50 C0    LDA   $C050
8061-   AD 57 C0    LDA   $C057
8064-   AD 52 C0    LDA   $C052
8067-   AD 10 C0    LDA   $C010
806A-   20 00 80    JSR   $8000
806D-   AD 00 C0    LDA   $C000
8070-   10 F5       BPL   $8067
8072-   60          RTS
```

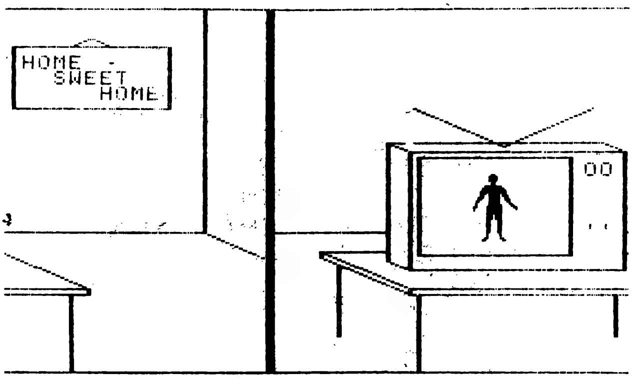


图 2.46 运行中的一个画面

弹出至各行0列，循环调用便实现图象不断向右移动显示。改动后见程序GP 2611，其程序原理就不再详细分析了。

程序GP 2611清单。

```
8000-   A0 27      LDY   #$27
8002-   A2 BF      LDX   #$BF
8004-   BD 00 81   LDA   $8100,X
8007-   85 07      STA   $07
8009-   BD 00 82   LDA   $8200,X
800C-   85 06      STA   $06
800E-   B1 06      LDA   ($06),Y
8010-   48         PHA
8011-   CA         DEX
8012-   E0 FF      CPX   #$FF
8014-   D0 EE      BNE   $8004
8016-   A2 00      LDX   #$00
8018-   BD 00 81   LDA   $8100,X
801B-   85 07      STA   $07
801D-   BD 00 82   LDA   $8200,X
8020-   85 06      STA   $06
8022-   C0 00      CPY   #$00
8024-   F0 10      BEQ   $8036
8026-   88         DEY
8027-   B1 06      LDA   ($06),Y
8029-   C8         INY
802A-   91 06      STA   ($06),Y
802C-   E8         INX
802D-   E0 C0      CPX   #$C0
802F-   D0 E7      BNE   $8018
8031-   88         DEY
8032-   C0 FF      CPY   #$FF
8034-   D0 E0      BNE   $8016
8036-   68         PLA
8037-   91 06      STA   ($06),Y
```

```
8039-    E8          INX
803A-    E0 C0       CPX    #$C0
803C-    90 DA       BCC    $8018
803E-    60          RTS
803F-    A9 20       LDA    #$20
8041-    85 E6       STA    $E6
8043-    A2 00       LDX    #$00
8045-    8A          TXA
8046-    48          PHA
8047-    20 11 F4    JSR    $F411
804A-    68          PLA
804B-    AA          TAX
804C-    A5 26       LDA    $26
804E-    9D 00 82    STA    $8200,X
8051-    A5 27       LDA    $27
8053-    9D 00 81    STA    $8100,X
8056-    E8          INX
8057-    E0 C0       CPX    #$C0
8059-    D0 EA       BNE    $8045
805B-    AD 54 C0    LDA    $C054
805E-    AD 50 C0    LDA    $C050
8061-    AD 57 C0    LDA    $C057
8064-    AD 52 C0    LDA    $C052
8067-    AD 10 C0    LDA    $C010
806A-    20 00 80    JSR    $8000
806D-    AD 00 C0    LDA    $C000
8070-    10 F5       BPL    $8067
8072-    60          RTS
```

第七章　移动清屏

本章主要介绍高分辨图形在移动过程中不断清屏的几个机器语言子程序，这和第六章介绍的移动显示程序有相似之处，共同点就是都能控制图象移动（上、下移动或左、右移动），而不同的是一个在不断显示，一个在显示过程中逐渐清屏。

为了学习和应用，本章中还就高分辨图形的其它方式清屏技巧，作了详细的介绍，这些技巧神态各异，既丰富多采，又生动活泼。这样，可以使非常单调刻板的显示方式变得饶有情趣，直接移植本章中的任何一个清屏技巧，到您编制的图形软件中，将会为您的软件增加光采。

第一节　向上、向下移动清屏

程序 GP2701为向上移动清屏子程序，程序 GP2702为向下移动清屏子程序，程序 GP2703为另外一种编程的向下移动清屏子程序。细心的读者如果仔细阅读一下 GP2701和 GP2702就可以发现，清屏和显示程序十分相似，所不同的就在于清屏是要消除图像，因而在图象不断向上、向下移动时，需要在最末行全部置 0，这样就达到了清屏的目的。因此，这里不再详细分析 GP2701和 GP2702子程序，请读者对照 GP2603和 GP2608可以了解程序编制的原理。

GP2703采用了另一种编程方法，实现由上向下移动清

屏，现在，分析一下它的原理：

$8000—$8026　　计算图象各行首地址。

其中：

$8000—$8003　　计数单元$1F置0，从0开始计算地址。

$8004—$8009　　根据图象存储地址的规律计算地址。左移2次，与#$1C相与，取$4、$8、$C、$10存入$07。

$800A—$8017　　再取$1F内容，右移4次，与#$03相与，取$1、$2丢掉其它，然后加上存入$07单元的内容．再

程序GP2701清单。

```
8000-    A2 BF        LDX    #$BF
8002-    BD 00 81     LDA    $8100,X
8005-    85 07        STA    $07
8007-    BD 00 82     LDA    $8200,X
800A-    85 06        STA    $06
800C-    CA           DEX
800D-    BD 00 81     LDA    $8100,X
8010-    85 09        STA    $09
8012-    BD 00 82     LDA    $8200,X
8015-    85 08        STA    $08
8017-    A0 27        LDY    #$27
8019-    B1 08        LDA    ($08),Y
801B-    91 06        STA    ($06),Y
801D-    88           DEY
801E-    10 F9        BPL    $8019
8020-    E0 FF        CPX    #$FF
8022-    D0 DE        BNE    $8002
8024-    A9 00        LDA    #$00
8026-    A0 00        LDY    #$00
8028-    91 06        STA    ($06),Y
```

```
802A-   C8          INY
802B-   C0 28       CPY   #$28
802D-   90 F9       BCC   $8028
802F-   60          RTS
8030-   A9 20       LDA   #$20
8032-   85 E6       STA   $E6
8034-   A2 00       LDX   #$00
8036-   8A          TXA
8037-   48          PHA
8038-   20 11 F4    JSR   $F411
803B-   68          PLA
803C-   AA          TAX
803D-   A5 26       LDA   $26
803F-   9D 00 82    STA   $8200,X
8042-   A5 27       LDA   $27
8044-   9D 00 81    STA   $8100,X
8047-   E8          INX
8048-   E0 C0       CPX   #$C0
804A-   D0 EA       BNE   $8036
804C-   AD 54 C0    LDA   $C054
804F-   AD 50 C0    LDA   $C050
8052-   AD 57 C0    LDA   $C057
8055-   AD 52 C0    LDA   $C052
8058-   AD 10 C0    LDA   $C010
805B-   20 00 80    JSR   $8000
805E-   AD 00 C0    LDA   $C000
8061-   10 F5       BPL   $8058
8063-   60          RTS
```

程序 GP2702清单。

```
8000-   A2 BF       LDX   #$BF
8002-   BD 00 81    LDA   $8100,X
8005-   85 07       STA   $07
8007-   BD 00 82    LDA   $8200,X
800A-   85 06       STA   $06
```

```
800C-   CA          DEX
800D-   BD 00 81    LDA   $8100,X
8010-   85 09       STA   $09
8012-   BD 00 82    LDA   $8200,X
8015-   85 08       STA   $08
8017-   A0 27       LDY   #$27
8019-   B1 08       LDA   ($08),Y
801B-   91 06       STA   ($06),Y
801D-   88          DEY
801E-   10 F9       BPL   $8019
8020-   E0 FF       CPX   #$FF
8022-   D0 DE       BNE   $8002
8024-   A9 00       LDA   #$00
8026-   A0 00       LDY   #$00
8028-   91 06       STA   ($06),Y
802A-   C8          INY
802B-   C0 28       CPY   #$28
802D-   90 F9       BCC   $8028
802F-   60          RTS
8030-   A9 20       LDA   #$20
8032-   85 E6       STA   $E6
8034-   A2 00       LDX   #$00
8036-   8A          TXA
8037-   48          PHA
8038-   20 11 F4    JSR   $F411
803B-   68          PLA
803C-   AA          TAX
803D-   A5 26       LDA   $26
803F-   9D 00 82    STA   $8200,X
8042-   A5 27       LDA   $27
8044-   9D 00 81    STA   $8100,X
8047-   E8          INX
8048-   E0 C0       CPX   #$C0
804A-   D0 EA       BNE   $8036
804C-   AD 54 C0    LDA   $C054
804F-   AD 50 C0    LDA   $C050
8052-   AD 57 C0    LDA   $C057
```

```
8055-    AD 52 C0      LDA   $C052
8058-    AD 10 C0      LDA   $C010
805B-    20 00 80      JSR   $8000
805E-    AD 00 C0      LDA   $C000
8061-    10 F5         BPL   $8058
8063-    60            RTS
```

程序 GP2703清单。

```
8000-    A9 00         LDA   #$00
8002-    85 1F         STA   $1F
8004-    0A            ASL
8005-    0A            ASL
8006-    29 1C         AND   #$1C
8008-    85 07         STA   $07
800A-    A5 1F         LDA   $1F
800C-    6A            ROR
800D-    6A            ROR
800E-    6A            ROR
800F-    6A            ROR
8010-    29 03         AND   #$03
8012-    05 07         ORA   $07
8014-    09 20         ORA   #$20
8016-    85 07         STA   $07
8018-    A5 1F         LDA   $1F
801A-    6A            ROR
801B-    29 E0         AND   #$E0
801D-    85 06         STA   $06
801F-    6A            ROR
8020-    6A            ROR
8021-    29 18         AND   #$18
8023-    05 06         ORA   $06
8025-    85 06         STA   $06
8027-    A9 00         LDA   #$00
8029-    A0 28         LDY   #$28
802B-    88            DEY
```

```
802C-    91 06       STA   ($06),Y
802E-    D0 FB       BNE   $802B
8030-    A9 5F       LDA   #$5F
8032-    20 A8 FC    JSR   $FCA8
8035-    18          CLC
8036-    A9 01       LDA   #$01
8038-    65 1F       ADC   $1F
803A-    85 1F       STA   $1F
803C-    C9 C0       CMP   #$C0
803E-    D0 C4       BNE   $8004
8040-    60          RTS
```

	加上＃$20，便得到高八位地址存入$07单元。
$8018—$801E	再取$1F单元内容，计算低八位地址，右移1次，与＃$E0相与，取$20、$40、$80、$A0、$C0送入$06单元。
$801F—$8026	继续右移2次，与＃$18相与，取$80、$10加上送入$06单元的内容便得到低八位地址送入$06单元。
$8027—$802F	将刚计算的这一行逐列置0清除。
$8030—$8034	延时，以便观察清屏的效果。
$8035—$803F	计数器加1，判断清屏是否结束，未清完转$8004继续下一行清屏。
$8040	清完返回。

演示GP2703子程序，可用程序GP2704。

```
10 D$ =  CHR$ (4)
20  PRINT D$"BLOAD PIC-1,A$2000"
30  PRINT D$"BLOAD GP2703"
40  POKE  - 16297,0: POKE  - 16300
    ,0
50  POKE  - 16302,0: POKE  - 16304
    ,0
60  FOR I = 1 TO 1000: NEXT I
70  CALL 32768
80  TEXT : HOME : END
```

第二节　向左、向右移动清屏

除了向左、向右移动显示外，有时还需要在图象移动过程中清屏，根据第六章介绍的原理，很容易编写向左、向右移动清屏子程序。GP 2705为向左移动清屏子程序， GP 2706为向右移动清屏子程序。因为清屏是要把图象清除，因此就不需要象显示那样将被移图象压栈保留，而只需要在被移行

程序GP 2705清单。

```
8000-    A0 00       LDY   #$00
8002-    A2 00       LDX   #$00
8004-    BD 00 81    LDA   $8100,X
8007-    85 07       STA   $07
8009-    BD 00 82    LDA   $8200,X
800C-    85 06       STA   $06
800E-    C0 27       CPY   #$27
8010-    F0 10       BEQ   $8022
8012-    C8          INY
8013-    B1 06       LDA   ($06),Y
8015-    88          DEY
8016-    91 06       STA   ($06),Y
8018-    E8          INX
```

```
8019-   E0 C0       CPX   #$C0
801B-   D0 E7       BNE   $8004
801D-   C8          INY
801E-   C0 28       CPY   #$28
8020-   D0 E0       BNE   $8002
8022-   A9 00       LDA   #$00
8024-   91 06       STA   ($06),Y
8026-   E8          INX
8027-   E0 C0       CPX   #$C0
8029-   90 D9       BCC   $8004
802B-   60          RTS
802C-   A9 20       LDA   #$20
802E-   85 E6       STA   $E6
8030-   A2 00       LDX   #$00
8032-   8A          TXA
8033-   48          PHA
8034-   20 11 F4    JSR   $F411
8037-   68          PLA
8038-   AA          TAX
8039-   A5 26       LDA   $26
803B-   9D 00 82    STA   $8200,X
803E-   A5 27       LDA   $27
8040-   9D 00 81    STA   $8100,X
8043-   E8          INX
8044-   E0 C0       CPX   #$C0
8046-   D0 EA       BNE   $8032
8048-   AD 54 C0    LDA   $C054
804B-   AD 50 C0    LDA   $C050
804E-   AD 57 C0    LDA   $C057
8051-   AD 52 C0    LDA   $C052
8054-   AD 10 C0    LDA   $C010
8057-   20 00 80    JSR   $8000
805A-   AD 00 C0    LDA   $C000
805D-   10 F5       BPL   $8054
805F-   60          RTS
```

程序GP2706清单。

```
8000-   A0 27      LDY  #$27
8002-   A2 00      LDX  #$00
8004-   BD 00 81   LDA  $8100,X
8007-   85 07      STA  $07
8009-   BD 00 82   LDA  $8200,X
800C-   85 06      STA  $06
800E-   C0 00      CPY  #$00
8010-   F0 10      BEQ  $8022
8012-   88         DEY
8013-   B1 06      LDA  ($06),Y
8015-   C8         INY
8016-   91 06      STA  ($06),Y
8018-   E8         INX
8019-   E0 C0      CPX  #$C0
801B-   D0 E7      BNE  $8004
801D-   88         DEY
801E-   C0 FF      CPY  #$FF
8020-   D0 E0      BNE  $8002
8022-   A9 00      LDA  #$00
8024-   91 06      STA  ($06),Y
8026-   E8         INX
8027-   E0 C0      CPX  #$C0
8029-   90 D9      BCC  $8004
802B-   60         RTS
802C-   A9 20      LDA  #$20
802E-   85 E6      STA  $E6
8030-   A2 00      LDX  #$00
8032-   8A         TXA
8033-   48         PHA
8034-   20 11 F4   JSR  $F411
8037-   68         PLA
8038-   AA         TAX
8039-   A5 26      LDA  $26
803B-   9D 00 82   STA  $8200,X
803E-   A5 27      LDA  $27
```

```
8040-    9D 00 81    STA   $8100,X
8043-    E8          INX
8044-    E0 C0       CPX   #$C0
8046-    D0 EA       BNE   $8032
8048-    AD 54 C0    LDA   $C054
804B-    AD 50 C0    LDA   $C050
804E-    AD 57 C0    LDA   $C057
8051-    AD 52 C0    LDA   $C052
8054-    AD 10 C0    LDA   $C010
8057-    20 00 80    JSR   $8000
805A-    AD 00 C0    LDA   $C000
805D-    10 F5       BPL   $8054
805F-    60          RTS
```

程序GP2707清单。

```
8000-    A0 00       LDY   #$00
8002-    A2 BF       LDX   #$BF
8004-    A9 00       LDA   #$00
8006-    85 1F       STA   $1F
8008-    A5 1F       LDA   $1F
800A-    0A          ASL
800B-    0A          ASL
800C-    29 1C       AND   #$1C
800E-    85 07       STA   $07
8010-    A5 1F       LDA   $1F
8012-    6A          ROR
8013-    6A          ROR
8014-    6A          ROR
8015-    6A          ROR
8016-    29 03       AND   #$03
8018-    05 07       ORA   $07
801A-    09 20       ORA   #$20
801C-    85 07       STA   $07
801E-    A5 1F       LDA   $1F
8020-    6A          ROR
8021-    29 E0       AND   #$E0
```

```
8023-    85 06         STA   $06
8025-    6A            ROR
8026-    6A            ROR
8027-    29 18         AND   #$18
8029-    05 06         ORA   $06
802B-    85 06         STA   $06
802D-    A9 00         LDA   #$00
802F-    91 06         STA   ($06),Y
8031-    18            CLC
8032-    A9 01         LDA   #$01
8034-    65 1F         ADC   $1F
8036-    85 1F         STA   $1F
8038-    CA            DEX
8039-    D0 CD         BNE   $8008
803B-    A9 5F         LDA   #$5F
803D-    20 A8 FC      JSR   $FCA8
8040-    C8            INY
8041-    98            TYA
8042-    C9 28         CMP   #$28
8044-    D0 BC         BNE   $8002
8046-    60            RTS
```

末列不断置0，除掉图象。而向左移动清屏或向右移动清屏，则只要改变列数，即从0列开始，逐列递增移动就是向左，从$27列开始，逐列递减移动就是向右。这两个程序我们不再详细分析了。

将程序GP2703略为改动一下，使其变为从左向右移动清屏，见程序GP2707。其原理请读者参照GP 2703自行分析，这里不再重复。

第三节 画线清屏技巧

高分辨图形的清屏方法较多，这里再介绍一个较为新颖

有趣的清屏程序：它可以将高分辨图形由上至下像演戏中拉幕一样清除，虽然说不上耳目一新，但可以使您的圆形软件有所增色。

例如，在高分辨第一页面存有一个名叫PIC1的图形，运行程序GP2708后，可将PIC1图形由上到下逐渐清屏，酷似拉戏幕一样，见图2.47。

程序GP2708清单。

```
10 D$ =  CHR$ (4)
20  PRINT D$;"BLOAD PIC 1"
25  POKE  - 16304,0
30  POKE  - 16297,0
35  POKE  - 16302,0
40  POKE  - 16300,0
50  FOR I = 1 TO 500: NEXT
100  GOSUB 190
150  FOR I = 1 TO 500: NEXT : TEXT
     : HOME : END
190  POKE  - 16300,0
200  HCOLOR= 3: FOR I = 0 TO 190
210  HPLOT 0,I TO 279,I: NEXT I
220  HCOLOR= 0: RETURN
```

原图形PIC1，见图2.47。

运行程序GP2708，瞬间PIC1的图形，见图2.48。

应该特别指出的是，程序GP2708的运行方法比较特殊，具体方法如下：

- 将程序GP2708键入内存。
- 改动190句为：190 HGR。
- 从100句开始运行程序，即GOTO100↙。
- 用上述步骤运行程序后，改190句为：190 POKE-

16300，0。

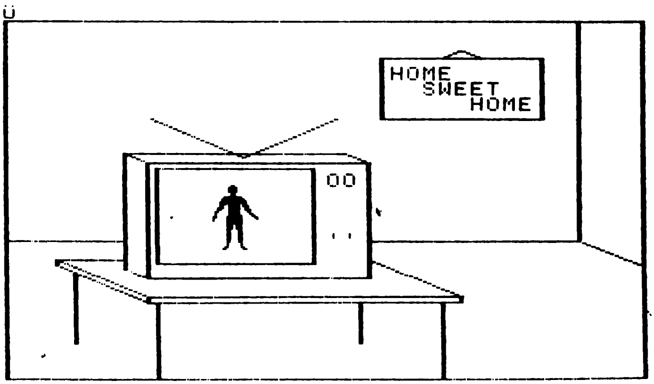


图 2.47

ÜPOKE1913,1

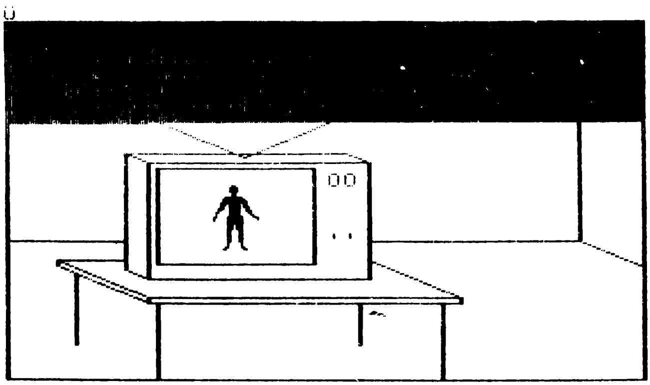


图 2.48

·重头运行程序GP2708，可以看到清屏过程。

又如，在高分辨第二页面存有一个名叫MLW的图形，

运行程序GP 2709后，也可将ML W图形，由上到下逐行清除。

程序GP 2709清单。

```
10 D$ =  CHR$ (4)
20  PRINT D$;"BLOAD MLW"
25  POKE  - 16304,0
30  POKE    16302,0
35  POKE  - 16297,0
40  POKE    16299,0
50  FOR I = 1 TO 500: NEXT
100  GOSUB 190
150  FOR I = 1 TO 500: NEXT   TEXT
     : HOME : END
190  POKE    16299,0
200  HCOLOR= 3: FOR I = 0 TO 190
210  HPLOT 0,I TO 279,I: NEXT
220  HCOLOR= 0: RETURN
```

原图形ML W，见图2.49。

UPOKE1913,1

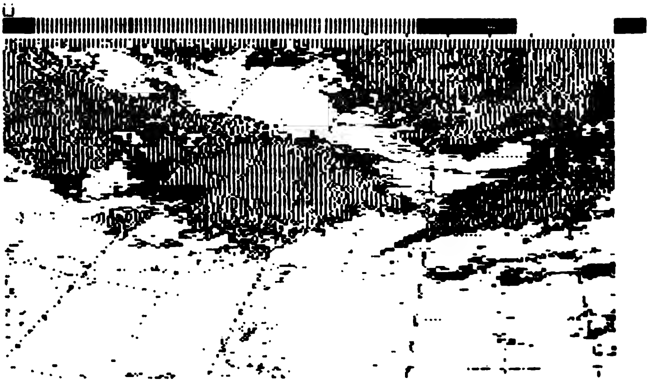


图 2.49

运行程序GP 2709，瞬间MLW的图形，见图2.50。

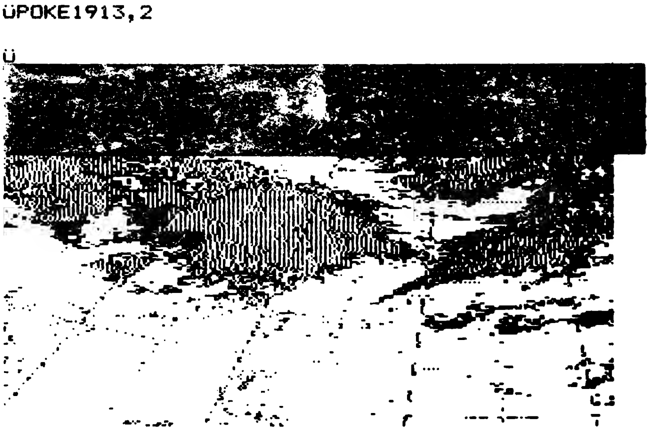


图 2.50

同上原理，程序GP 2709的运行步骤如下：

· 将程序GP 2709键入内存。

· 改动190句为：190HGR 2。

· GOTO100 ↙。

· 改190句为：190POKE-16299,0。

· 重新运行程序GP 2709。

程序GP 2708和GP 2709设计思想和程序结构是十分相近的，唯不同的是软开关设置有些差别。此外，图形PIC1存贮在高分辨第一页；图形MLW存贮在高分辨率第二页，在使用时应注意这些差别。

为简化上述操作，并使程序更为简洁，程序GP 2708改

为GP 2710。

```
10  HGR : POKE  - 16302,0
15 D$ =  CHR$ (4)
20  PRINT D$;"BLOAD PIC 1"
25  HCOLOR= 3: FOR I = 0 TO 190
30  HPLOT 0,I TO 279,I: NEXT
32  HCOLOR= 0
35  FOR I = 1 TO 500: NEXT : TEXT
     : HOME : END
```

同理，程序GP 2709改为GP 2711。

```
10  HGR2
20 D$ =  CHR$ (4)
30  PRINT D$;"BLOAD MLW"
40  FOR I = 1 TO 500: NEXT
50  HCOLOR= 3: FOR I = 0 TO 190
60  HPLOT 0,I TO 279,I: NEXT
70  HCOLOR= 0
80  FOR I = 1 TO 500: NEXT : TEXT
     : HOME : END
```

程序GP 2710和程序GP 2711比较简单但实用，它们都是采用循环的方法，控制高分辨绘图语句采用画线的手段清屏的。这就给我们一个启发，巧用循环语句，例如改变循环变量的初值、终值、步长，采用几重循环；或者变化绘线语句HPLOT的参数和格式，将使高分辨图形的清屏更加多样生动，进而使图形软件更加丰富多采。为此，这里提供一个具有18种不同方式的清屏程序。其操作方法，由于采用总控的方式而显得特别简单，程序运行后，该图形PIC 1显示完成后，键入1—18中任一个数字并↙，就是一种清屏方式，

而要结束操作键入0即可。

程序共有18种不同方式的清屏：

① 从上到下。
② 从下到上。
③ 从右到左。
④ 从左到右。
⑤ 从中间向左、右两边。
⑥ 从两边向中间。
⑦ 从中间向上、下。
⑧ 从上、下同时向中间。
⑨ 从四角向中间。
⑩ 从中间向四角
⑪ 从上到下拉百叶窗。
⑫ 从左到右拉百叶窗。
⑬ 从左上、右下同时反方向。
⑭ 从左上、右下到中间，然后从右上、左下向中间。
⑮ 从左到右，从上到下双向拉百叶窗。
⑯ 从左到右，从上到下变化图案。
⑰ 从左到右暗条纹。
⑱ 从右到左暗条纹。

完整的程序GP 2712清单。

```
10  HGR : POKE  - 16302,0
20 D$ =  CHR$ (4)
30  PRINT D$;"BLOAD PIC 1"
40 K$ = "4000<2000.3FFFM N D823G"
50  FOR I = 1 TO  LEN (K$): POKE 5
      11 + I, ASC ( MID$ (K$,I,1))
```

```
     128: NEXT I: POKE 72,0
60  CALL  - 144
70  INPUT "X= ";X
80  IF X = 0 THEN  TEXT : HOME : END

90  ON X GOSUB 130,180,200,250,280
     ,330,350,400,450,470,490,560,
     630,680,770,890,1020,1090
100  GOSUB 1160: CALL  - 144
110  POKE  - 16304,0: POKE  - 1929
     7,0: POKE  - 16302,0: POKE  -
     16300,0
120  GOTO 70
130  FOR I = 1 TO 500: NEXT
140  HCOLOR= 3: FOR I = 0 TO 190
150  HPLOT 0,I TO 279,I: NEXT
160  HCOLOR= 0
170  FOR I = 1 TO 500: NEXT : TEXT
     : HOME : RETURN
180  HCOLOR= 3
182  FOR I = 190 TO 0 STEP  - 2
184  HPLOT 0,I TO 279,I: NEXT
186  HCOLOR= 0
190  FOR I = 1 TO 500: NEXT : TEXT
     : HOME : RETURN
200  FOR I = 1 TO 500: NEXT
210  HCOLOR= 3
215  FOR I = 279 TO 0 STEP  - 1
220  HPLOT I,0 TO I,190: NEXT
230  HCOLOR= 0
240  FOR I = 1 TO 500: NEXT : TEXT
     : HOME : RETURN
250  HCOLOR= 3
255  FOR I = 140 TO 0 STEP  - 1
260  HCOLOR= 3
262  FOR I = 0 TO 279
264  HPLOT I,0 TO I,190: NEXT
266  HCOLOR= 0
```

```
270  FOR I = 1 TO 500: NEXT : TEXT
     : HOME : RETURN
280  FOR I = 1 TO 500: NEXT
290  HCOLOR= 3
295  FOR I = 140 TO 0 STEP - 1
300  HPLOT I,0 TO I,190: HPLOT 279
      - I,0 TO 279 - I,190: NEXT
310  HCOLOR= 0
320  FOR I = 1 TO 500: NEXT : TEXT
     : HOME : RETURN
330  HCOLOR= 3
332  FOR I = 0 TO 140
334  HPLOT I,0 TO I,190
336  HPLOT 279 - I,0 TO 279 - I,19
     0
338  NEXT : HCOLOR= 0
340  FOR I = 1 TO 500: NEXT : TEXT
     : HOME : RETURN
350  FOR I = 1 TO 500: NEXT
360  HCOLOR= 3: FOR I = 0 TO 95
370  HPLOT 0,95 - I TO 279,95 - I:
      HPLOT 0,95 + I TO 279,95 + I
     : NEXT
380  HCOLOR= 0
390  FOR I = 1 TO 500: NEXT : TEXT
     : HOME : RETURN
400  FOR I = 1 TO 500: NEXT
410  HCOLOR= 3: FOR I = 0 TO 95
420  HPLOT 0,I TO 279,I: HPLOT 0,1
     90 - I TO 279,190 - I: NEXT
430  HCOLOR= 0
440  FOR I = 1 TO 500: NEXT : TEXT
     : HOME : RETURN
450  HCOLOR= 3
454  FOR I = 0 TO 140
456  HPLOT I,I TO 279 - I,I TO 279
      - I,190 - I TO I,190 - I TO
```

```
     I,I
458  NEXT : HCOLOR= 0
460  FOR I = 1 TO 500: NEXT : TEXT
     : HOME : RETURN
470  HCOLOR= 3
474  FOR I = 140 TO 0 STEP  - 1
476  HPLOT I,I TO 279 - I,I TO 279
     - I,190 - I TO I,190 - I TO
     I,I
478  NEXT : HCOLOR= 0
480  FOR I = 1 TO 500: NEXT : TEXT
     : HOME : RETURN
490  HCOLOR= 3: FOR I = 0 TO 32
500  FOR J = 0 TO 160 STEP 32
510  IF J = 160 THEN J = J - 1
520  HPLOT 0,I + J TO 279,I + J
530  NEXT J: NEXT I
540  HCOLOR= 0
550  FOR I = 1 TO 500: NEXT : TEXT
     : HOME : RETURN
560  HCOLOR= 3: FOR I = 0 TO 35
570  FOR J = 0 TO 245 STEP 35
580  IF J = 245 THEN J = J - 1
590  HPLOT I + J,0 TO I + J,191
600  NEXT J: NEXT I
610  HCOLOR= 0
620  FOR I = 1 TO 500: NEXT : TEXT
     : HOME : RETURN
630  HCOLOR= 3: FOR I = 0 TO 191
640  HPLOT 0,I TO 140,I
650  HPLOT 141,191 - I TO 279,191 -
     I
660  NEXT I: HCOLOR= 0
670  FOR I = 1 TO 500: NEXT : TEXT
     : HOME : RETURN
680  HCOLOR= 3: FOR I = 0 TO 95
690  HPLOT 0,I TO 140,I
```

```
700  HPLOT 141,191 - I TO 279,191 -
      I
710  NEXT I
720  FOR I = 0 TO 95
730  HPLOT 0,191 - I TO 141,191 -
      I
740  HPLOT 141,I TO 279,I
750  NEXT I: HCOLOR= 0
760  FOR I = 1 TO 500: NEXT : TEXT
      : HOME : RETURN
770  HCOLOR= 3: FOR I = 0 TO 17
780  FOR J = 0 TO 245 STEP 35
790  IF J = 245 THEN J = J - 1
800  HPLOT I + J,0 TO I + J,191
810  NEXT J: NEXT I
820  FOR I = 0 TO 16
830  FOR J = 0 TO 160 STEP 32
840  IF J = 160 THEN J = J - 1
850  HPLOT 0,I + J TO 279,I + J
860  NEXT J: NEXT I
870  HCOLOR= 0
880  FOR I = 1 TO 500: NEXT : TEXT
      : HOME : RETURN
890  HCOLOR= 3: FOR I = 0 TO 35
900  FOR J = 0 TO 255 STEP 35
910  IF J > 244 THEN J = 244
920  HPLOT I + J,0 TO I + J,191
930  HPLOT I + J,0 TO I + J,191
940  HCOLOR= 0:K = J -  INT (J / 3
      5) * 3
950  IF K >  = 160 THEN K = 159
960  IF I > 32 AND K    = 159 THEN
      980
970  HPLOT 0,I + K TO 279,I + K
980  HCOLOR= 3
990  NEXT J: NEXT I
1000  HPLOT 0,0 TO 0,191 TO 279,19
```

```
     1
1010  FOR I = 1 TO 5500: NEXT : TEXT
     : HOME : RETURN
1020  HCOLOR= 0: FOR I = 0 TO 35
1030  FOR J = 0 TO 245 STEP 35
1040  IF J = 245 THEN J = J - 1
1050  HPLOT I + J,0 TO I + J,191
1060  NEXT J: NEXT I
1070  HCOLOR= 0
1080  FOR I = 1 TO 500: NEXT : TEXT
     : HOME : RETURN
1090  HCOLOR= 0
1095  FOR I = 35 TO 0 STEP  - 1
1100  FOR J = 245 TO 0 STEP  - 35
1110  IF J = 245 THEN J = J + 1
1120  HPLOT I + J,0 TO I + J,191
1130  NEXT J: NEXT I
1140  HCOLOR= 0
1150  FOR I = 1 TO 500: NEXT   TEXT
     : HOME : RETURN
1160 K$ = "2000<4000.5FFFM N D823G
     "
1170  FOR I = 1 TO  LEN (K$): POKE
     511 + I, ASC ( MID$ (K$,I,1))
      + 128: NEXT I: POKE 72,0
1180  RETURN
```

第八章 趣味显示

在第六章和第七章，我们比较详细地介绍了一些最基本的图形移动显示与图形移动清屏子程序，搞清楚这些程序的原理，并在此基础上稍作一些改动，就可以变化出各种形式的显示和清屏程序来。本章主要介绍拉幕显示与清屏，合幕显示与清屏，上下拼合显示与清屏等几个机器语言子程序，

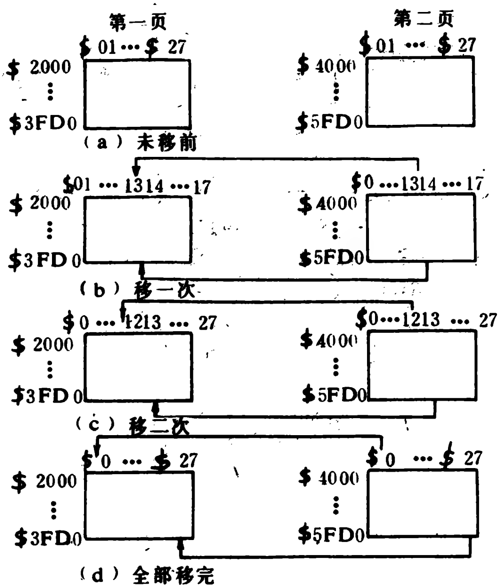


图 2.51

并剖析其原理。同前面几章一样，这里介绍的程序，都可以移植，为您所用，可以编制一些BASIC程序，调用您的图形，但要注意机器语言的入口地址，用CALL命令选择正确的调用点。

第一节 拉幕显示与清屏

舞台演出时，大幕拉开，演出开始了。GP 2801子程序也有这样的效果。其原理见图2.51将图像存放于第二页，然后将中间两列分别调往第一页中间两列显示，再向左右列分别减增，持续不断调往第一页显示，直至全部显完。

程序GP 2801清单。

```
8000-   A9 14      LDA   #$14
8002-   8D 1E 80   STA   $801E
8005-   A9 13      LDA   #$13
8007-   8D 24 80   STA   $8024
800A-   A2 00      LDX   #$00
800C-   BD 00 81   LDA   $8100,X
800F-   85 07      STA   $07
8011-   18         CLC
8012-   69 20      ADC   #$20
8014-   85 09      STA   $09
8016-   BD 00 82   LDA   $8200,X
8019-   85 06      STA   $06
801B-   85 08      STA   $08
801D-   A0 0E      LDY   #$0E
801F-   B1 08      LDA   ($08),Y
8021-   91 06      STA   ($06),Y
8023-   A0 19      LDY   #$19
8025-   B1 08      LDA   ($08),Y
8027-   91 06      STA   ($06),Y
8029-   E8         INX
802A-   E0 C0      CPX   #$C0
```

```
802C-    D0 DE       BNE    $800C
802E-    A9 A0       LDA    #$A0
8030-    20 A8 FC    JSR    $FCA8
8033-    CE 1E 30    DEC    $801E
8036-    EE 24 80    INC    $8024
8039-    AD 24 80    LDA    $8024
803C-    C9 28       CMP    #$28
803E-    D0 CA       BNE    $800A
8040-    60          RTS
8041-    A9 20       LDA    #$20
8043-    85 E6       STA    $E6
8045-    A2 00       LDX    #$00
8047-    8A          TXA
8048-    48          PHA
8049-    20 11 F4    JSR    $F411
804C-    68          PLA
804D-    AA          TAX
804E-    A5 26       LDA    $26
8050-    9D 00 82    STA    $8200,X
8053-    A5 27       LDA    $27
8055-    9D 00 81    STA    $8100,X
8058-    E8          INX
8059-    E0 C0       CPX    #$C0
805B-    D0 EA       BNE    $8047
805D-    AD 54 C0    LDA    $C054
8060-    AD 50 C0    LDA    $C050
8063-    AD 57 C0    LDA    $C057
8066-    AD 52 C0    LDA    $C052
8069-    AD 10 C0    LDA    $C010
806C-    20 00 80    JSR    $8000
806F-    AD 00 C0    LDA    $C000
8072-    10 F5       BPL    $8069
8074-    60          RTS
```

程序GP 2801原理说明如下：

$8000—$8009	初始化，设置列指针，指向屏幕中央。
$800A—$801C	查表，分别取一、二页行地址。
$801D—$8028	传送中间两列开始拉幕显示，见图2.51(b)。
$8029—$802D	行+1，判图象各行是否显示完，未完转$800C继续。
$802E—$8032	延时，增强拉幕显示效果。
$8033—$803F	列向左边递减，右边递增，形成拉幕，然后判是否全部移完，见图2.51(c),2.51(d)。未完转$800A继续。
$8040	移完结束返回。
$8041—$805C	同前面程序，计算各行首地址。
$805D—$8074	演示用。

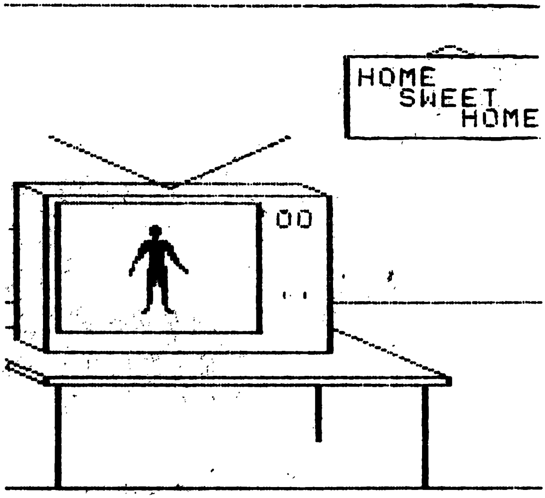
演示程序,GP2801，可用程序GP2802。

程序GP2802清单。

```
10 D$ =  CHR$ (4)
20  PRINT D$"BLOAD PIC-1,A$4000"
30  PRINT D$"BLOAD GP2801"
40  HGR : CALL 32833
50  TEXT : HOME : END
```

程序运行中的一个画面，图2.52。

拉幕显示要将图象存于第二页，而拉幕清屏则要图象在第一页显示，然后从中间两列开始向两边移动，不断置0，清除图象，达到拉幕清屏的效果，见程序GP2803，演示要注意先显示图象，再执行程序。



KE1913,1

图 2.52

程序GP2803清单

```
8000-    A9 14        LDA   #$14
8002-    8D 17 80     STA   $8017
8005-    A9 13        LDA   #$13
8007-    8D 1D 80     STA   $801D
800A-    A2 00        LDX   #$00
800C-    BD 00 81     LDA   $8100,X
800F-    85 07        STA   $07
8011-    BD 00 82     LDA   $8200,X
8014-    85 06        STA   $06
8016-    A0 05        LDY   #$05
8018-    A9 00        LDA   #$00
801A-    91 06        STA   ($06),Y
801C-    A0 22        LDY   #$22
```

```
801E-   91 06      STA   ($06),Y
8020-   E8         INX
8021-   E0 C0      CPX   #$C0
8023-   D0 E7      BNE   $800C
8025-   A9 A0      LDA   #$A0
8027-   20 A8 FC   JSR   $FCA8
802A-   CE 17 80   DEC   $8017
802D-   EE 1D 80   INC   $801D
8030-   AD 1D 80   LDA   $801D
8033-   C9 28      CMP   #$28
8035-   D0 D3      BNE   $800A
8037-   60         RTS
8038-   A9 20      LDA   #$20
803A-   85 E6      STA   $E6
803C-   A2 00      LDX   #$00
803E-   8A         TXA
803F-   48         PHA
8040-   20 11 F4   JSR   $F411
8043-   68         PLA
8044-   AA         TAX
8045-   A5 26      LDA   $26
8047-   9D 00 82   STA   $8200,X
804A-   A5 27      LDA   $27
804C-   9D 00 81   STA   $8100,X
804F-   E8         INX
8050-   E0 C0      CPX   #$C0
8052-   D0 EA      BNE   $803E
8054-   AD 54 C0   LDA   $C054
8057-   AD 50 C0   LDA   $C050
805A-   AD 57 C0   LDA   $C057
805D-   AD 52 C0   LDA   $C052
8060-   AD 10 C0   LDA   $C010
8063-   20 00 80   JSR   $8000
8066-   AD 00 C0   LDA   $C000
8069-   10 F5      BPL   $8060
806B-   60         RTS
```

第二节　合幕显示与清屏

合幕显示也是将图像存放于第二页，但与拉幕显示相反，它是从两边向中间慢慢显示，它的原理见图2.53和程序GP 2804。

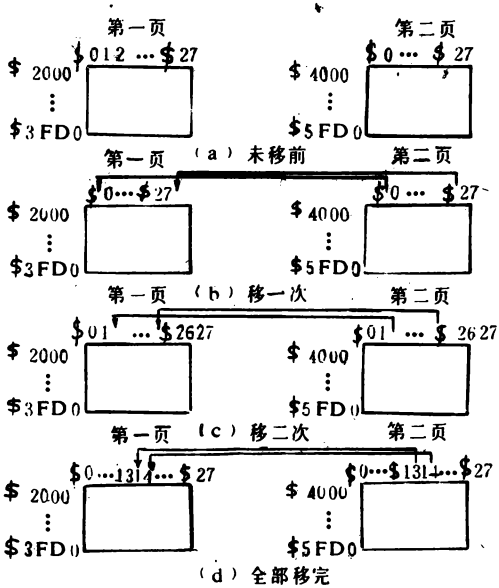


图 2.53

对照程序GP 2801，可见，程序GP 2804实际上只改动了几个字节，但显示形式完全不同。同理还可以再改动个别字节，将变幻出更多的显示形式，请读者自己去思考。

对合幕清屏，也与上面拉幕清屏相似，只要改动程序GP 2803中的几个字节，见GP 2805。演示GP 2804与GP

程序GP2804清单

```
8000-   A9 00       LDA  #$00
8002-   8D 1E 80    STA  $801E
8005-   A9 27       LDA  #$27
8007-   8D 24 80    STA  $8024
800A-   A2 00       LDX  #$00
800C-   BD 00 81    LDA  $8100,X
800F-   85 07       STA  $07
8011-   18          CLC
8012-   69 20       ADC  #$20
8014-   85 09       STA  $09
8016-   BD 00 82    LDA  $8200,X
8019-   85 06       STA  $06
801B-   85 08       STA  $08
801D-   A0 07       LDY  #$07
801F-   B1 08       LDA  ($08),Y
8021-   91 06       STA  ($06),Y
8023-   A0 20       LDY  #$20
8025-   B1 08       LDA  ($08),Y
8027-   91 06       STA  ($06),Y
8029-   E8          INX
802A-   E0 C0       CPX  #$C0
802C-   D0 DE       BNE  $800C
802E-   A9 A0       LDA  #$A0
8030-   20 A8 FC    JSR  $FCA8
8033-   EE 1E 80    INC  $801E
8036-   CE 24 80    DEC  $8024
8039-   AD 24 80    LDA  $8024
803C-   C9 13       CMP  #$13
803E-   D0 CA       BNE  $800A
8040-   60          RTS
8041-   A9 20       LDA  #$20
8043-   85 E6       STA  $E6
8045-   A2 00       LDX  #$00
8047-   8A          TXA
8048-   48          PHA
8049-   20 11 F4    JSR  $F411
```

```
804C-   68          PLA
804D-   AA          TAX
804E-   A5 26       LDA   $26
8050-   9D 00 82    STA   $8200,X
8053-   A5 27       LDA   $27
8055-   9D 00 81    STA   $8100,X
8058-   E8          INX
8059-   E0 C0       CPX   #$C0
805B-   D0 EA       BNE   $8047
805D-   AD 54 C0    LDA   $C054
8060-   AD 50 C0    LDA   $C050
8063-   AD 57 C0    LDA   $C057
8066-   AD 52 C0    LDA   $C052
8069-   AD 10 C0    LDA   $C010
806C-   20 00 80    JSR   $8000
806F-   AD 00 C0    LDA   $C000
8072-   10 F5       BPL   $8069
8074-   60          RTS
```

程序GP 2805清单。

```
8000-   A9 00       LDA   #$00
8002-   8D 17 80    STA   $8017
8005-   A9 27       LDA   #$27
8007-   8D 1D 80    STA   $801D
800A-   A2 00       LDX   #$00
800C-   BD 00 81    LDA   $8100,
800F-   85 07       STA   $07
8011-   BD 00 82    LDA   $8200,X
8014-   85 06       STA   $06
8016-   A0 0D       LDY   #$0D
8018-   A9 00       LDA   #$00
801A-   91 06       STA   ($06),Y
801C-   A0 1A       LDY   #$1A
801E-   91 06       STA   ($06),Y
```

```
8020-   E8          INX
8021-   E0 C0       CPX   #$C0
8023-   D0 E7       BNE   $800C
8025-   A9 A0       LDA   #$A0
8027-   20 A8 FC    JSR   $FCA8
802A-   EE 17 80    INC   $8017
802D-   CE 1D 80    DEC   $801D
8030-   AD 1D 80    LDA   $801D
8033-   C9 13       CMP   #$13
8035-   D0 D3       BNE   $800A
8037-   60          RTS
8038-   A9 20       LDA   #$20
803A-   85 E6       STA   $E6
803C-   A2 00       LDX   #$00
803E-   8A          TXA
803F-   48          PHA
8040-   20 11 F4    JSR   $F411
8043-   68          PLA
8044-   AA          TAX
8045-   A5 26       LDA   $26
8047-   9D 00 82    STA   $8200,X
804A-   A5 27       LDA   $27
804C-   9D 00 81    STA   $8100,X
804F-   E8          INX
8050-   E0 C0       CPX   #$C0
8052-   D0 EA       BNE   $803E
8054-   AD 54 C0    LDA   $C054
8057-   AD 50 C0    LDA   $C050
805A-   AD 57 C0    LDA   $C057
805D-   AD 52 C0    LDA   $C052
8060-   AD 10 C0    LDA   $C010
8063-   20 00 80    JSR   $8000
8066-   AD 00 C0    LDA   $C000
8069-   10 F5       BPL   $8060
806B-   60          RTS
```

2805，可参照演示GP 2801与GP 2803的方法。图2.54为合幕显示程序运行中的一个画面。

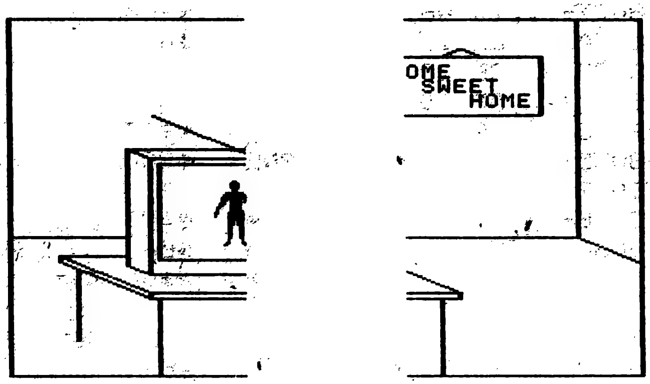


图 2.54

第三节 上下拼合显示与清屏

在图象上下移动显示程序的基础上稍加改进，即可实现图象上下拼合显示。将图象存放于第二页，然后将图象分两半分别调往第一页从上下向中间移动，直至图象合拢，就形成了上下拼合显示。其原理见程序GP 2806和见图2.55。

程序GP 2806原理说明如下：

$8000—$8007　　初始化，将图象中间位置指针放在$1E和$1F单元。

程序GP 2804清单。

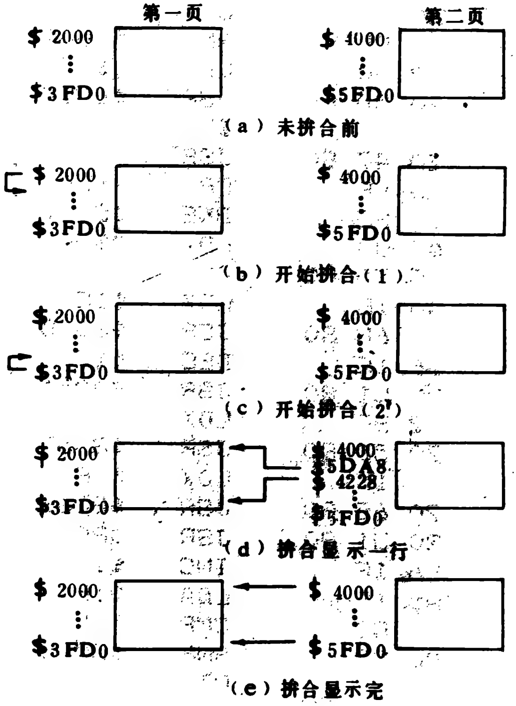


图 2.55

程序GP2806清单。

```
8000-   A9 5F      LDA   #$5F
8002-   85 1E      STA   $1E
8004-   A9 60      LDA   #$60
8006-   85 1F      STA   $1F
8008-   A2 60      LDX   #$60
800A-   20 5D 80   JSR   $805D
800D-   CA         DEX
```

```
800E-   20 47 80    JSR   $8047
8011-   E0 00       CPX   #$00
8013-   D0 F5       BNE   $800A
8015-   A2 60       LDX   #$60
8017-   20 5D 80    JSR   $805D
801A-   E8          INX
801B-   20 47 80    JSR   $8047
801E-   E0 BF       CPX   #$BF
8020-   D0 F5       BNE   $8017
8022-   A2 00       LDX   #$00
8024-   20 5D 80    JSR   $805D
8027-   A6 1E       LDX   $1E
8029-   20 68 80    JSR   $8068
802C-   C6 1E       DEC   $1E
802E-   20 51 80    JSR   $8051
8031-   A2 BF       LDX   #$BF
8033-   20 5D 80    JSR   $805D
8036-   A6 1F       LDX   $1F
8038-   20 68 80    JSR   $8068
803B-   20 51 80    JSR   $8051
803E-   E6 1F       INC   $1F
8040-   A5 1F       LDA   $1F
8042-   C9 C0       CMP   #$C0
8044-   D0 C2       BNE   $8008
8046-   60          RTS
8047-   BD 00 82    LDA   $8200,X
804A-   85 08       STA   $08
804C-   BD 00 81    LDA   $8100,X
804F-   85 09       STA   $09
8051-   A0 00       LDY   #$00
8053-   B1 08       LDA   ($08),Y
8055-   91 06       STA   ($06),Y
8057-   C8          INY
8058-   C0 28       CPY   #$28
805A-   D0 F7       BNE   $8053
805C-   60          RTS
```

```
805D-   BD 00 81    LDA  $8100,X
8060-   85 07       STA  $07
8062-   BD 00 82    LDA  $8200,X
8065-   85 06       STA  $06
8067-   60          RTS
8068-   BD 00 82    LDA  $8200,X
806B-   85 08       STA  $08
806D-   BD 00 81    LDA  $8100,X
8070-   18          CLC
8071-   69 20       ADC  #$20
8073-   85 09       STA  $09
8075-   60          RTS
8076-   A9 20       LDA  #$20
8078-   85 E6       STA  $E6
807A-   A2 00       LDX  #$00
807C-   8A          TXA
807D-   48          PHA
807E-   20 11 F4    JSR  $F411
8081-   68          PLA
8082-   AA          TAX
8083-   A5 26       LDA  $26
8085-   9D 00 82    STA  $8200,X
8088-   A5 27       LDA  $27
808A-   9D 00 81    STA  $8100,X
808D-   E8          INX
808E-   E0 C0       CPX  #$C0
8090-   D0 EA       BNE  $807C
8092-   AD 54 C0    LDA  $C054
8095-   AD 50 C0    LDA  $C050
8098-   AD 57 C0    LDA  $C057
809B-   AD 52 C0    LDA  $C052
809E-   AD 10 C0    LDA  $C010
80A1-   20 00 80    JSR  $8000
80A4-   60          RTS
```

$8008—$800C　　取图象中间位置地址放于$06.

	＄07单元中，作为目的行地址。
＄800D－＄8010	取图象另一中间位置的地址放于＄08,＄09单元中，作为被移行地址,并由上向下移一行，见图2.55(b)。
＄8011－＄8014	判断上半幅是否移完，未完转＄800A继续移。
＄8015－＄801D	移完，再将下半幅由下向上移动，见图2.55(c)。
＄801E－＄8021	判断下半幅是否移完，未完转＄8017继续移。
＄8022－＄8026	移完，取第一页首行作为目的行地址。
＄8027－＄802B	取第二页中间位置地址作为被移行地址。
＄802C——＄802D	＄1E减1,为下一行作好准备。
＄802E－＄8030	调移动一行子程序移动一行,见图2.55(d)。
＄8031－＄8035	取第一页末行作为目的行地址。
＄8036－＄803A	取第二页另一中间位置地址作为被移行地址。
＄803B－＄803D	调移动一行子程序移动一行，见图2.55(d)。
＄803E－＄8045	＄1F加1，兼作计数用，判断是否拼合完,未完转＄8008继续，见图2.55(e)。

$ 8046	拼合完返回结束。
$ 8047—$ 805C	取目的行地址，并移动一行子程序。

其中：

$ 8047—$ 8050	查表，取地址放入$ 08，$ 09单元中，作为目的行地址。
$ 8051—$ 805C	移动一行子程序。
$ 805D—$ 8067	查表，取地址放入$ 06，$ 07单元中，作为被移行地址子程序。
$ 8068—$ 8075	查表，将高八位加$ 20，取第二页地址作为目的行地址放入$ 08，$ 09单元子程序。
$ 8076—$ 8091	计算各行首地址。
$ 8092—$ 80A4	演示用。

演示程序GP2802，只需改动一下即可。图2.56为正在

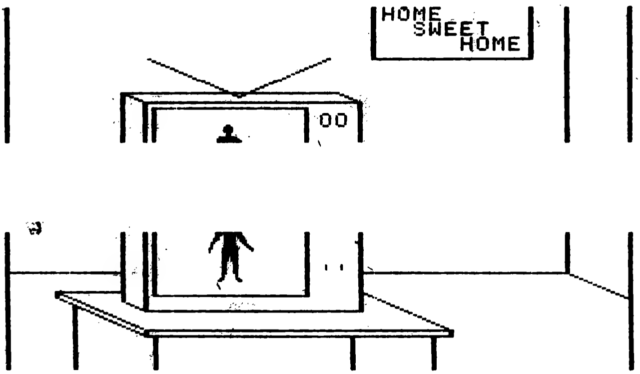


图 2.56

拼合显示的画面。

上下拼合清屏，见程序GP2807，其程序原理说明如下：

$8000—$8009　　初始化，设置首行，末行指针。

$800A—$800E　　调查表取地址，请一行子程序将首行清屏。

```
8000-    A9 00       LDA   #$00
8002-    8D 0B 80    STA   $800B
8005-    A9 BF       LDA   #$BF
8007-    8D 10 80    STA   $8010
800A-    A2 60       LDX   #$60
800C-    20 22 80    JSR   $8022
800F-    A2 5F       LDX   #$5F
8011-    20 22 80    JSR   $8022
8014-    EE 0B 80    INC   $800B
8017-    CE 10 80    DEC   $8010
801A-    AD 10 80    LDA   $8010
801D-    C9 5F       CMP   #$5F
801F-    D0 E9       BNE   $800A
8021-    60          RTS
8022-    BD 00 82    LDA   $8200,X
8025-    85 06       STA   $06
8027-    BD 00 81    LDA   $8100,X
802A-    85 07       STA   $07
802C-    A0 00       LDY   #$00
802E-    A9 00       LDA   #$00
8030-    91 06       STA   ($06),Y
8032-    C8          INY
8033-    C0 28       CPY   #$28
8035-    D0 F7       BNE   $802E
8037-    A9 A0       LDA   #$A0
8039-    20 A8 FC    JSR   $FCA8
803C-    60          RTS
803D-    A9 20       LDA   #$20
```

```
803F-   85 E6      STA   $E6
8041-   A2 00      LDX   #$00
8043-   8A         TXA
8044-   48         PHA
8045-   20 11 F4   JSR   $F411
8048-   68         PLA
8049-   AA         TAX
804A-   A5 26      LDA   $26
804C-   9D 00 82   STA   $8200,X
804F-   A5 27      LDA   $27
8051-   9D 00 81   STA   $8100,X
8054-   E8         INX
8055-   E0 C0      CPX   #$C0
8057-   D0 EA      BNE   $8043
8059-   AD 54 C0   LDA   $C054
805C-   AD 50 C0   LDA   $C050
805F-   AD 57 C0   LDA   $C057
8062-   AD 52 C0   LDA   $C052
8065-   20 00 80   JSR   $8000
8068-   60         RTS
```

\$800F—\$8013	调子程序将末行清屏。
\$8014—\$8020	首行加1，末行减1，判是否清完，未完转\$800A继续。
\$8021	清完返回结束。
\$8022—\$802B	查表，取地址放入\$06，\$07单元中，作为目的行地址。
\$802C—\$8036	清一行。
\$8037—\$803B	延时用。
\$803C	子程序返回。
\$803D—\$8058	计算各行首地址。
\$8059—\$8068	演示用。

第九章 窗口剪辑

本章介绍在图形管理中十分有用的图形剪辑技术。首先讲述“窗口”和“视见区”的概念，然后通过对折线剪辑的实例分析，详细说明改变窗口和视见区对图形的影响，所有这些，都是信息绘图中十分重要的，又是最基本的。在本章的后面两节，着重介绍图形剪辑和图形编辑工具软件使用方法，这样，您可以对图形进行移位、放大、反相、颠倒、翻身等各种处理，从而设计出形形式式、千姿百态的图形来。

第一节 窗口技术

如果我们对一幅图象中的某个特定区域的画面发生兴趣，以便根据需要把它放大，仔细研究其细节，或者突出它的部位，可以用一定的方法，将其周围的画面全部擦除。

画面上某个特定区域，可以用人为的方法指定，即设置一个矩形“窗口”。保留窗口内的画面，而擦除窗口外的画面，就叫图形剪辑。

在我们介绍的程序中，用户选择的工作窗口，按下述方法定义：

W1：最小横坐标。

W2：最大横坐标。

W3：最小纵坐标：

W4 最大纵坐标。

显然，窗口越大，原图象留下来的多，剪去的少，在相同视见区（见后面解释）情况下，局部图象变小。反之，窗口越小，原图象留下来的少，剪去的多，在相同视见区情况下，局部图象得到了放大。

当画面上定义了一个窗口并进行裁剪以后，常需要把窗口内的画面部分在屏幕上指定区域内显示出来，以便达到放大或缩小的目的。屏幕上用于显示窗内画面的区域，叫做视见区。

显然，在同样窗口的情况下，视见区越大，画面愈大；视见区越小，画面愈小。

从这里我们可以看到窗口与视见区的差别：窗口用来指出在屏幕上要看到的是些什么，而视见区则用来指出在屏幕何处显示其内容。视见区通常也是一个矩形区域，也由人们自由地确定。它可以看到复制的窗口内容。

下面的程序中，视见区用V1，V2，V3，V4作为变量：

V1：最小横坐标。

V2：最大横坐标。

V3：最小纵坐标。

V4：最大纵坐标。

在下面一节，我们将通过几个实例，介绍图形剪辑的方法。

第二节　折线剪辑

题目是将DATA语句中的数据画成折线图（见250—260句）。

程序GP2901是没有调用画视见区矩形框的子程序（见

63500—63510句）时，画出折线的程序。

```
üPOKE1913,2

ü
```



图 2.57

程序G P 2901清单。

```
130  TEXT : HOME
140 W1 = 0:W2 = 279:W3 = 0:W4 = 19
    1
150  HGR2 : HCOLOR= 3
155  READ V1,V2,V3,V4
157  DATA      0 ,279, 0,191
160  GOSUB 60000
170 N = 11
180  READ X,Y
190 W$ = "U": GOSUB 62300
200  FOR I = 2 TO N
210  READ X,Y
220 W$ = "D": GOSUB 62300
230  NEXT I
240  END
245  DATA    27,35
```

```
250   DATA 27,35,33,20,45,60,56,80,
      77,40,89,58,110,120,130,29,15
      0,160
260   DATA   270,0
```

运行结果见图2.57。

程序G P 2902调用了63500开始的一段子程序(画视见区方框),并将窗口减小，图形放大了，见图2.58。

程序G P 2902清单。

```
130   TEXT : HOME
140 W1 = 0:W2 = 100:W3 = 0:W4 = 80
150   HGR2 : HCOLOR= 3
155   READ V1,V2,V3,V4
157   DATA        0 ,279, 0,191
160   GOSUB 60000
165   GOSUB 63500
170 N = 11
180   READ X,Y
190 W$ = "U": GOSUB 62300
200   FOR I = 2 TO N
210   READ X,Y
220 W$ = "D": GOSUB 62300
230   NEXT I
240   END
245   DATA     27,35
250   DATA 27,35,33,20,45,60,56,80,
      77,40,89,58,110,120,130,29,15
      0,160
260   DATA   270,0
```

运行结果见图2.58。

```
ÜPOKE1913,2
```

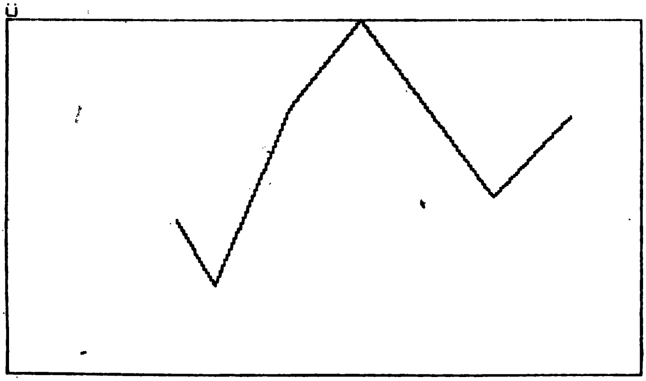


图 2.58

程序GP2903和程序GP2902相比，窗口加大，图形缩小（视见区不变，见图2.59）。

程序GP2903清单。

```
130  TEXT : HOME
140 W1 = 0:W2 = 180:W3 = 0:W4 = 14
     0
150  HGR2 : HCOLOR= 3
155  READ V1,V2,V3,V4
157  DATA       0 ,279, 0,191
160  GOSUB 60000
165  GOSUB 63500
170 N = 11
180  READ X,Y
190 W$ = "U": GOSUB 62300
200  FOR I = 2 TO N
210  READ X,Y
220 W$ = "D": GOSUB 62300
230  NEXT I
```

```
240   END
245   DATA    27,35
250   DATA 27,35,33,20,45,60,56,80,
      77,40,89,58,110,120,130,29,15
      0,160
260   DATA   270,0
```

运行结果见图2.59。

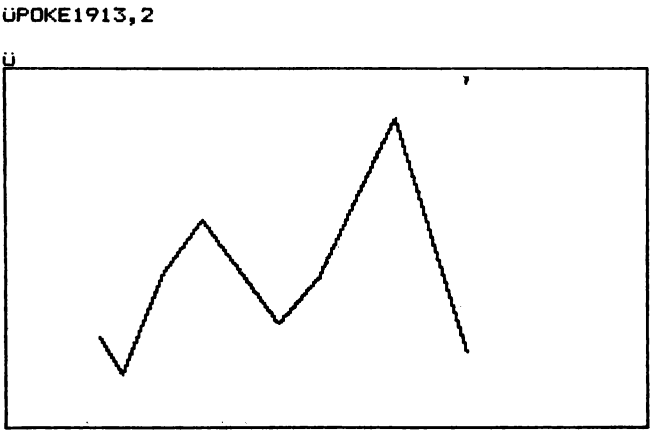


图 2.59

程序GP2904和程序GP2903不同之处，在于程序GP2904视见区减小，因而整个图形缩小（窗口相同，见图2.60）。

程序GP2904清单。

```
130   TEXT    HOME
140 W1 = 0:W2 = 180:W3 = 0:W4 = 14
     0
```

```
150  HGR2 : HCOLOR= 3
155  READ V1,V2,V3,V4
157  DATA  0,152,0,175
160  GOSUB 60000
165  GOSUB 63500
170 N = 11
180  READ X,Y
190 W$ = "U": GOSUB 62300
200  FOR I = 2 TO N
210  READ X,Y
220 W$ = "D": GOSUB 62300
230  NEXT I
240  END
245  DATA    27,35
250  DATA 27,35,33,20,45,60,56,80,
     77,40,89,58,110,120,130,29,15
     0,160
260  DATA   270,0
```

运行结果见图2.60。

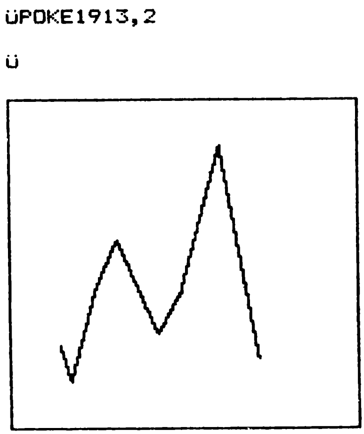


图 2.60

以上4个程序（GP2901，GP2902，GP2903，GP2904）都要用到下面几个子程序：

① 完成用户坐标转换到屏幕或绘图机坐标（60000—60010句）

```
60000 A8 = (V2 - V1) / (W2 - W1):B
      8 = (V1 * W2 - V2 * W1) / (W2
       - W1)
60010 A9 = (V4 - V3) / (W4 - W3):B
      9 = (V3 * W4 - V4 * W3) / (W4
       - W3): RETURN
```

②完成坐标变换并画线（62300—62400句）

```
62300 X6 = X:Y6 = Y: REM   ....ENT
      RY FOR QUICK TRACE NO CUTTING

62400 X5 = A8 * X6 + B8:Y5 = 191 -
      (A9 * Y6 + B9)
62410  IF W$ = "U" THEN  HPLOT X5,
      Y5: RETURN
62420  HPLOT  TO X5,Y5: RETURN
```

③ 完成画视见区的矩形框（63500—63510）

```
63500 X6 = W1:Y6 = W3:W$ = "U": GOSUB
      62400:X6 = W2:W$ = "D": GOSUB
      62400
63510 Y6 = W4: GOSUB 62400:X6 = W1
      : GOSUB 62400:Y6 = W3: GOSUB
      62400: RETURN
```

值得指出的是上述修剪现象，在信息绘图中是很重要的，又是基本的。

第三节 图形剪辑

对已有图形资料进行剪辑、修改、放缩、合并处理，从而产生新的更为丰富多采的画面，是一件饶有情趣的事。利用剪辑程序GP2905，可以节省大量的作图时间，并使新产生的图象达到更为夸张的效果。

例如，有一幅小人画像，经本程序处理后放大了，见图2.61和图2.62。

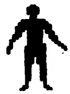


图 2.61 原小人图象

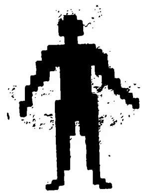


图 2.62 放大后的小人图象

程序GP2905清单。

```
5   HOME :S = 32768:X = 0:Y = 0
10   POKE  - 16297,0: POKE  - 16300
      ,0: POKE  - 16302,0: POKE  -
      16304,0
15   POKE 230,32
20   FOR I = 0 TO 9: READ A: POKE S
      + I,A: NEXT
30   DATA 1,0,4,0,45,245,27,54,6,0
40   HCOLOR= 3: ROT= 0: SCALE= 1: POKE
      232,0: POKE 233,128
50 K =  PEEK ( - 16384): IF K > 12
      8 THEN 70
60   XDRAW 1 AT X,Y: FOR T = 1 TO 1
```

```
     5: NEXT T
65  XDRAW 1 AT X,Y: GOTO 50
70  POKE  - 16368,0: IF K <  > 155
      THEN 90
80 D = D + 1:X(D) = X:Y(D) = Y: IF
     D = 2 THEN 110
85 X = X + 3:Y = Y + 3: ROT= 32
90 X = X + (K = 203) - (K = 202)
95 Y = Y + (K = 205) - (K = 201)
100  GOTO 50
110  POKE  - 16301,0:D = 0:X = 0:Y
      = 0: ROT= 0
120  VTAB 21: INPUT "A,B=";A,B
130  INPUT "X,Y=";X,Y:XX = X
140  DIM A(A * B * 2)
150  FOR I = 1 TO  INT (B + 0.99)
160  IF A <  = 1 THEN 200
170  FOR J = 1 TO  INT (A + 0.99) -
     1
180 N = N + 1:A(N) = 5 + 2 * D
190  NEXT J
200 N = N + 1:A(N) = 6:D =  NOT D
210  NEXT I:S = S + 4
220  FOR I = 1 TO N STEP 2
230  POKE S,A(I + 1) * 8 + A(I):S =
     S + 1
240  NEXT I: POKE S,0
250  POKE 230,64: POKE  - 16302,0:
      POKE  - 16299,0
260 N =  INT (X(1) / 7):C = X(1) -
     N * 7
270  FOR I = Y(1) TO Y(2)
280 F =  INT (I / 8):G =  INT (F /
     8):H = F - 8 * G:K = I - F *
     8
290 S = 8192 + 1024 * K + 128 * H +
     40 * G + N:J = X(1)
```

```
300  IF C = 0 THEN 330
310 R =  PEEK (S) / 2 ^ C
320  FOR D = C TO 6: GOTO 340
330 R =  PEEK (S): FOR D = 0 TO 6
340 M =  INT (R / 2):P = R - M * 2
     :R = M
350  IF J > X(2) THEN X = XX: GOTO
     390
360  POKE 228,127 * P: DRAW 1 AT X
     ,Y
370 J = J + 1:X = X + A: NEXT D
380 S = S + 1: GOTO 330
390 Y = Y + B: NEXT I
400  POKE - 16368,0: IF  PEEK ( -
     16384) < 128 THEN 400
410  TEXT : HOME : END
```

程序GP 2905能对指定的一个矩形区域内的图形进行放大处理（局部放大），也可以对一幅完整画面进行变形处理（变宽或变窄），使之达到更为夸张的效果。

使用方法：

·将两幅图形分别读入第一页和第二页图形显示区。例如BLOAD P1,A$2000↙,BLOAD P2 A$4000↙。

·运行上述程序，屏幕上出现第一页图形（P1），并在屏幕左上角显示一个“┌”形状的闪动光标，此时可按I,J,K，M键控制光标作上、左、右、下4个方向移动，在移到需要放大处理的那一部分图形左上角时，按“ESC”键，光标立即变成“┘”形状，再按动上述4键，至需要处理的图形右下方，重按“ESC”键，即可确定剪辑的图形范围。

·输入横向放大系数A和纵向放大系数B，并输入起始坐标X、Y。这时即可看到被剪辑处理的图象由起始位置X、

Y处逐步显示出来（在第二页）。

·图形显示完成后，按任意键结束。用 BSAVE 文件名，A$ 4000，L$ 2000存盘，以后使用时，用BLOAD文件名，A$ 4000调入内存，加显示程序段，即可看到图形P1处理剪辑过的图形合并在P2图形上。

·若只对第一页图形作单一放缩处理，请改动 250 句为HGR2，360句改为：IF P =1 THEN DRAW1 AT X，Y。这样可以加快显示速度。

第四节 图形编辑

存放在 $6000—$618E 单元中的机器语言程序，是一个可以对整幅图形实行编辑的工具软件，文件名为 GP 2906，其特点是简洁、方便。

将图形编辑程序 GP 2906装入内存后，再将所需编辑图形调至高分辨率第一页，执行 CALL 24576（BASIC 状

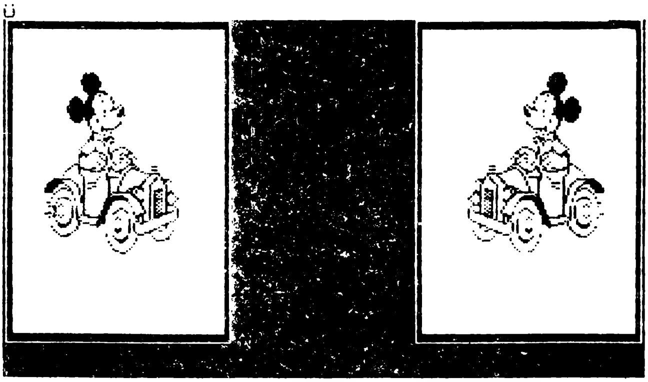


图 2.63

态）或6000G ↵（监控状态下），就可以对图形进行编辑加工。

按"D"键使图形垂直下移；按"L"键使图形水平左移；按"Y"键使图形左右翻身；按"X"键使图形上下颠倒；按"C"键使图形反相；按"E"键则结束运行。

下面是一个运行实例，编辑前的图形只有一个米老鼠坐在汽车上利用程序GP 2906水平移动和左右翻身的功能，画画上出现两幅互为镜像的米老鼠图案，见图2.63。

程序GP 2906清单。

```
6000- AD 50 C0 AD 52 C0 AD 54
6008- C0 AD 57 C0 A0 00 84 3A
6010- 84 3C A0 40 84 3B C8 84
6018- 3D A0 BF 98 0A 0A 29 1C
6020- 85 15 98 6A 6A 6A 6A 29
6028- 03 05 15 09 20 91 3C 98
6030- 6A 29 E0 85 14 6A 6A 29
6038- 18 05 14 91 3A 88 C0 FF
6040- D0 D9 2C 00 C0 10 FB AD
6048- 00 C0 C9 C4 D0 3C A0 28
6050- 88 B9 00 20 99 00 42 98
6058- D0 F6 A2 01 BD 00 40 85
6060- 14 BD 00 41 85 15 A0 28
6068- 88 B1 14 85 3A B9 00 42
6070- 91 14 A5 3A 99 00 42 98
6078- D0 EE 8A F0 0A E8 E0 C0
6080- D0 DA A2 00 4C 5C 60 4C
6088- 42 60 C9 CC D0 28 A2 C0
6090- CA BD 00 40 85 14 BD 00
6098- 41 85 15 A0 00 B1 14 85
60A0- 3A C8 B1 14 88 91 14 C8
60A8- C0 27 D0 F5 A5 3A 91 14
60B0- 8A D0 DD 4C 42 60 2C 10
```

```
60B8- C0 C9 D9 D0 5B A0 00 84
60C0- 11 A4 11 B9 00 40 85 14
60C8- B9 00 41 85 15 A2 27 86
60D0- 3E A0 00 A2 13 84 3A A4
60D8- 3A B1 14 85 3B 85 3C A4
60E0- 3E B1 14 85 3D 85 3F A0
60E8- 08 06 3B 6A 88 D0 FA 06
60F0- 3C 6A A4 3E 91 14 A0 08
60F8- 06 3D 6A 88 D0 FA 06 3F
6100- 6A A4 3A 91 14 C6 3E E6
6108- 3A E4 3E D0 CA E6 11 A2
6110- C0 E4 11 D0 AC 4C 42 60
6118- C9 D8 D0 40 A2 60 A0 00
6120- 84 11 A0 BF 84 12 A4 11
6128- B9 00 40 85 14 B9 00 41
6130- 85 15 A4 12 B9 00 40 85
6138- 16 B9 00 41 85 17 A0 00
6140- B1 14 85 3A B1 16 91 14
6148- A5 3A 91 16 C8 C0 28 D0
6150- EF E6 11 C6 12 E4 11 D0
6158- CD 4C 42 60 C9 C3 D0 2A
6160- A0 20 A2 40 AD 00 20 0A
6168- 49 FF 6A 8D 00 20 EE 65
6170- 61 EE 6C 61 D0 EE EE 66
6178- 61 EE 6D 61 EC 66 61 D0
6180- E3 8C 66 61 8C 6D 61 4C
6188- 42 60 C9 C5 D0 F9 60
```

程序GP 2908是一个功能较强的图形编辑程序，采用BASIC语言编程。其中调用两个机器语言子程序，一个就是上面介绍的GP2906程序（$6000—$618E），另一个是GP 2907程序（$300—$3BF），后者可以对调进内存的图形进行反相、翻身、颠倒等九项功能处理，前者则对调进内存的图形进行垂直下移、水平左移等四项功能处理，配合使

用这两个程序，将会设计出形态多样，更会夸张的图形来。

运行程序GP 2908后，机器自动调进程序GP 2907和程序GP2906进内存，然后给出“INPUT PICT NAME:”的提示，这时操作者输入任意一个已存于盘片中的图形名，并按回车，此时磁盘机动作，稍等片刻后，图形即自动调进内存，然后按数字键1—9中任意一个数字，图形变换，形态各异。当您满意某一个已变换好的图形时，再按数字0，进入编辑状态，按“D”键使图形下移，按“L”键使图形左移，配合使用这两个键，使图形处于屏幕中的合适位置，然后按“E”键结束，处理好的图形可以存盘，也可以打印出来。

程序GP 2907清单：

```
0300- 20 A5 03 20 25 03 A9 28
0308- 85 FC 85 EE A9 4F 85 FE
0310- 85 EF 20 25 03 A9 50 85
0318- FC 85 EE A9 77 85 FE 85
0320- EF 20 25 03 60 A0 00 B1
0328- FC 85 FB A2 07 46 FB 26
0330- FA CA D0 F9 A9 7F 25 FA
0338- 85 FA 84 EB 38 A5 FE 85
0340- E3 E5 EB 85 FE A0 00 A5
0348- FA 91 FE A5 E3 85 FE A4
0350- EB C8 C0 14 D0 D1 A5 FD
0358- 18 69 04 85 FD A5 FF 18
0360- 69 04 85 FF C9 40 90 BD
0368- E6 EC A5 EC C9 02 F0 19
0370- A5 FC 18 69 80 85 FC A5
0378- FE 18 69 80 85 FE A5 ED
0380- 85 FD A5 28 85 FF 4C 25
0388- 03 A0 00 84 EC A5 EE 85
0390- FC A5 EF 85 FE E6 ED A5
0398- ED 85 FD E6 28 A5 28 85
03A0- FF C9 24 90 80 A9 00 85
```

```
03A8- EC 85 FC 85 EE A9 20 85
03B0- ED 85 FD A9 20 85 28 85
03B8- FF A9 27 85 FE 85 EF 60
```

程序GP 2908清单：

```
5  REM   GP2908
6 D$ =  CHR$ (4)
7  PRINT D$;"BLOAD GP2907"
8  PRINT D$;"BLOAD GP2906"
10  INPUT "INPUT PICT NAME:";K$
30  PRINT D$"BLOAD"K$",A$2000
50  HOME
60  POKE  - 16304,0: POKE  - 16297
     ,0: POKE  - 16300,0
70  POKE  - 16301,0: POKE 230,32
80 A$ = "A0ÜEXIT A1ÜHALF PAGE NOR-
     SYMM COPY A2ÜHALF PAGE INV-SY
     MM COPY A3ÜHALF PAGE IOR-SAME
      COPY A4ÜHALF PAGE INV-SAME C
     OPY A5ÜALL PAGE NOR-SYMM COPY
      A6ÜALL PAGE INV-SYMM COPY A7
     ÜALL PAGE NOR-SAME COPY A8ÜAL
     L PAGE INV-SAME COPY A9ÜREBLO
     AD"
90  HTAB 1: VTAB 23: PRINT  LEFT$
     (A$,39);:A$ =  MID$ (A$,2) +
      LEFT$ (A$,1)
95 Q =  PEEK ( - 16384): IF Q < 12
     8 THEN  FOR Q = 1 TO 100: NEXT
     :Q =  FRE (0): GOTO 90
100  POKE  - 16368,0:Q = Q - 176: IF
     Q < 0 OR Q > 9 THEN 90
110  IF Q = 0 THEN 300
120  IF Q = 9 THEN  PRINT : GOTO 5
     0
```

```
130  FOR R = 1 TO  INT (Q / 2 + 0.
     5)
140  READ A,B,C,D,E,F,G,H,I,J,K,L,
     M,N,O,P
150  POKE 781,A: POKE 796,B: POKE
     810,C: POKE 811,D
160  POKE 812,E: POKE 813,F: POKE
     814,G: POKE 833,H
170  POKE 834,I: POKE 835,J: POKE
     836,K: POKE 851,L
180  POKE 869,M: POKE 930,N: POKE
     948,O: POKE 954,P
190  NEXT R: RESTORE
200  IF  INT (Q / 2) = Q / 2 THEN
      POKE 822,69: GOTO 220
210  POKE 822,37
220  CALL 768
230  IF Q > 4 THEN  POKE  - 16299,
     0: POKE  - 16302,0: POKE 230,
     64: FOR S = 0 TO 2000: NEXT
240  GOTO 60
250  DATA 79,119,251,162,7,70,251,
     229,235,133,254,20,64,36,32,3
     9
260  DATA  60,100,250,76,52,3,234,
     76,71,3,234,20,64,36,32,20
270  DATA 79,119,251,162,7,70,251,
     229,235,133,254,40,96,68,64,3
     9
280  DATA   40,80,250,76,52,3,234,
     76,71,3,234,40,96,68,64,0
300  CALL 24576
```

第十章 造 型 动 画

本章首先简述动画设计的各种方法，然后分析汇编语言调用造型表的原理，接着介绍单个造型动画、多个造型动画显示方法和键盘操作控制，最后给出一个造型动画的设计实例。

文中提到的调用造型的机器语言子程序，多个造型图形数据都可直接引用，其程序结构、设计思想、编制原理都可以作为编制动画设计的借鉴。

第一节 简单动画显示技术

所谓动画显示技术，是指将屏幕上的图形，按照一定要求有规则地显示其变化或活动的过程。

如何形象逼真地模仿客观实体的活动过程，特别是有生命物体的活动情况，常常是游戏程序和教学辅助软件中十分需要的，因为一个连续运动的、真实感很强的动画，会给人们留下深刻的印象。

在动画显示技术中，动画对象的图形组成越复杂，程序处理的速度就越慢，而图形闪烁及呆滞现象也会随之产生。因此，减少画面的闪烁，使图形显示逼真，就是动画显示技术中的难点和关键。

可以用各种方法组织和设计动画页面。例如显示一幅图形，然后把它抹去，把这个图形进行变换，再予以显示，当

上述过程迅速地一幕接一幕进行时，在视觉上便产生运动。但随着组成图形的数据增多，实现变换的计算时间加大，图形变化的速度减慢，最终便失去了动态变化的连续性，这是一个缺点。另一种常用而有效的方法是，把图形的动态变化情况分成一幕幕，并按幕存贮起来，则先显示第一幕，接着抹去第一幕；显示第二幕，以后又抹去第二幕；显示下面一幕……，循环往复，同样也能实现图形的动态变化，优点是提高动态变化速度，但增加了存贮空间。再有一种实现图形变化的方法是采用背景变化的技巧，这在电影特技中经常使用。最后，也是我们这里主要介绍的方法是，利用中华学习机绘画显示的两个页面，将动画图形画在不同的页面上，然后控制屏幕页面的软开关，也能达到良好的动画显示效果。

作为一个实例，我们可以利用绘图工具软件，在高分辨显示的第一页面上画一个睁着眼睛的小熊猫(见图2.64)，而在第二页面上画一个闭着眼睛的小熊猫(见图2.65)，然后用程序GP2A01来实现小熊猫不断地眨眼。

程序GP2A01清单。

```
5  GOSUB 100
10  POKE  - 16304,0
20  POKE  - 16297,0
30  POKE  - 16302,0
40  POKE  - 16300,0
45  FOR J = 1 TO 50: NEXT J
50  POKE  - 16299,0
55  FOR P = 1 TO 50: NEXT P
60  GOTO 40
100 D$ =  CHR$ (4)
110  PRINT D$"BLOADP2-1,A$2000"
120  PRINT D$"BLOADP2-2,A$4000"
140  RETURN
```

第一页的熊猫图象，见图2.64。



图 2.64

第二页的熊猫图象，见图2.65。



图 2.65

程序GP 2A 01的软开关是用POKE指令设置的，也可以用调用相应的机器语言子程序GP2A02来完成。方法是在监控状态下按入机器语言子程序GP2A02，然后用程序GP2A03来实现熊猫眨眼的动态变化。

机器语言子程序GP2A02清单。

```
*300.319

0300- AD 50 C0 AD 57 C0 AD 52
0308- C0 AD 54 C0 A5 00 20 A8
0310- FC AD 55 C0 A5 00 20 A8
0318- FC 60
```

对应的汇编语言清单。

```
0300-   AD 50 C0    LDA   $C050
0303-   AD 57 C0    LDA   $C057
0306-   AD 52 C0    LDA   $C052
0309-   AD 54 C0    LDA   $C054
030C-   A5 00       LDA   $00
030E-   20 A8 FC    JSR   $FCA8
0311-   AD 55 C0    LDA   $C055
0314-   A5 00       LDA   $00
0316-   20 A8 FC    JSR   $FCA8
0319-   60          RTS
```

程序GP2A03清单。

```
5 S = 150
10 D$ =  CHR$ (4)
20  PRINT D$"BLOADP2-1,A$2000"
30  PRINT D$"BLOADP2-2,A$4000"
40  PRINT D$"BRUN GP2A02"
50  POKE 0,S
60  FOR J = 1 TO 20: CALL 768
70  NEXT : GOTO 50
```

GP2A04又是一个动画显示的实例，在这个程序中，利用了分页显示技术，图形搬家方法，以及调用左右易位处理机器语言子程序GP2402，结果在屏幕上看到的一个小熊猫

左右翻身同时摇头的显示，生动活泼，富有情趣。

将程序 GP 2 A04和程序GP 2402同存一盘中， RUN GP2 A04↙，即可看到活动的画面。

程序GP 2 A04清单。

```
2 D$ = CHR$ (4)
4  PRINT D$"BLOAD P2,A$2000"
5  PRINT D$"BLOADGP2402,A$300"
6  GOSUB 100: CALL  - 144
8  POKE 230,32: CALL 768
10  POKE  - 16304,0
20  POKE  - 16297,0
30  POKE  - 16302,0
40  POKE  - 16300,0
45  FOR J = 1 TO 500: NEXT J
50  POKE  - 16299,0
55  FOR P = 1 TO 500: NEXT P
60  GOTO 40
100 X$ = "4000<2000.3FFFM N D823G"

110  FOR I = 1 TO  LEN (X$): POKE
     511 + I, ASC ( MID$ (X$,I,1))
      + 128: NEXT I: POKE 72,0
120  RETURN
```

程序GP 2402清单。

```
*300.353
↵
0300- A9 00 A2 00 A0 00 85 EC
0308- 20 11 F4 A0 00 B1 26 20
0310- 3E 03 A5 EE 85 EF 98 AA
0318- 8E 1F 03 A9 27 38 E9 13
0320- A8 B1 26 20 3E 03 A5 EF
0328- 91 26 8A A8 A5 EE 91 26
0330- C8 C0 14 D0 D8 E6 EC A5
```

原图象见图2.66。

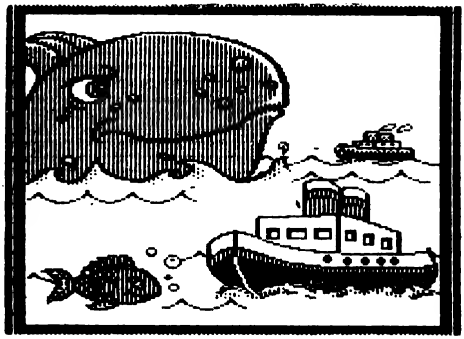


图 2.66

变换后图象见图2.67。

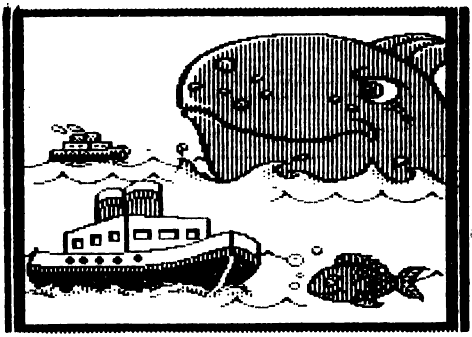


图 2.67

```
0338-  EC  C9  C0  D0  C5  60  86  ED
0340-  2A  26  EE  6A  A2  07  6A  26
0348-  EE  CA  E0  00  F0  03  4C  46
0350-  03  A6  ED  60
```

第二节　汇编语言调用造型表

编制一段汇编语言来调用造型表，不仅速度要比 BASIC 调用造型表快好几倍，而且用在动画设计上，动态效果突出且无闪烁感。

编制机器语言调用造型表，并不十分麻烦，只需要了解一些造型表的存贮单元和调用 ROM中一些有关造型的机器语言子程序，同时正确存贮造型表。

我们先给出调用造型表的机器语言子程序（$300—$32B），并给出对应的汇编语言程序（文件名统称GP2A05），然后逐段加以解释，从中可以了解到程序编制的方法。

程序GP2A05清单。

```
0300-  20 E2 F3 A9 00 85 E8 A9
0308-  60 85 E9 A2 03 20 F0 F6
0310-  A9 03 85 E7 A2 01 20 30
0318-  F7 A2 80 A0 00 A9 32 20
0320-  11 F4 A6 1A A4 1B A9 00
0328-  20 5D F6 60
```

程序GP2A05的汇编语言清单。

```
0300-   20 E2 F3     JSR   $F3E2
0303-   A9 00        LDA   #$00
0305-   85 E8        STA   $E8
0307-   A9 60        LDA   #$60
0309-   85 E9        STA   $E9
030B-   A2 03        LDX   #$03
030D-   20 F0 F6     JSR   $F6F0
0310-   A9 03        LDA   #$03
0312-   85 E7        STA   $E7
0314-   A2 01        LDX   #$01
```

```
0316-   20 30 F7    JSR   $F730
0319-   A2 80       LDX   #$80
031B-   A0 00       LDY   #$00
031D-   A9 32       LDA   #$32
031F-   20 11 F4    JSR   $F411
0322-   A6 1A       LDX   $1A
0324-   A4 1B       LDY   $1B
0326-   A9 00       LDA   #$00
0328-   20 5D F6    JSR   $F65D
032B-   60          RTS
```

一、高分辨图形显示页面的设置

同APPLESOFT一样，首先应该设置高分辨图形显示页面，例如，高分辨第一页作图页（相当于 HGR）的入口地址为＄F3E2，高分辨第二页作图页（相当于HGR2）的入口地址为＄F3D8，用机器语言调用，其指令分别是：

20 E2 F3 JSR ＄F3E2；启动第一页

20 D8 F3 JSR ＄F3D8；启动第二页

本例是启动高分辨第一页作图页面。

同BASIC一样，若要清除当前屏幕画面（用HOME指令），机器语言则应这样安排：

20 58 FC JSR ＄FC58

其中＄FC58为HOME指令的入口地址。本例没有安排。

二、造型表的地址

在BASIC程序中，＄E8（232）和＄E9（233）两个单元是专门用来存放造型表首地址的，并规定造型表首地址的低位（本例是＄00）放在＄E8中，而造型表首地址的高位（本例是＄60）放在＄E9中。BASIC程序中应安排：

POKE 232, 0: POKE233,96

对应的机器语言应为：

```
A9  00  LDA  #$00
85  E8  STA  $E8
A9  60  LDA  #$60
85  E9  STA  $E9
```

本例造型表地址放在$6000开始的单元中。只有这样安排，造型表才能送入内存并付诸使用。

三、高分辨作图的颜色

在BASIC程序中，通过HCOLOR指令设定高分辨作图的颜色，例如HCOLOR＝3。但在汇编语言中颜色代码（可取$00—$07中一个）应放在X暂存器中，并且必须通过调用程序名为HCOLOR的机器语言子程序来实现，它的入口地址是$F6F0。所以机器语言中颜色设定应由下面两句来实现：

```
A2  03          LDX  #$03
20  F0  F6      JSR  $F6F0
```

本例颜色代码为03。

四、绘图矢量的大小

在BASIC语言中，关于绘图矢量的大小是用SCALE＝N设定的，例如SCALE＝1表示造型表的放大倍数为1（不放大）。用机器语言处理上述指令时，要把造型放大的值放在$E7单元中，即

```
A9  03  LDA  #$03
85  E7  STA  $E7
```

本例造型放大值取03。

五、造型表显示位置（起点位置）的确定

在BASIC语言中如执行DRAW 1 AT140,94，表示将第一个造型绘于141列95行开始的地方。在机器语言中则要调用＄F411开始的一段机器语言子程序，并在调用前将显示起点位置水平坐标的低位放入X暂存器中，高位放入Y暂存器中，而造型显示起点位置的垂直坐标放在累加器A中，即有下面4条指令：

```
A2  80       LDX  #$80
A0  00       LDY  #$00
A9  32       LDA  #$32
20  11  F4   JSR  $F411
```

本例坐标选在X＝128，Y＝00这一点显示造型位置。

六、调用造型的指针设定

调用第几个造型，这也可以用调用机器语言子程序来实现，其入口地址在＄F730中，调用前应将造型序号（第几个造型）之值放入X暂存器中，即：

```
A2  01       LDX  #$01
20  30  F7   JSR  $F730
```

本例是调第一个造型，指针号为01。此值相当于BASIC中造型绘图命令DRAW（或XDRAW）后面紧跟的参数。

七、关于造型显示

在BASIC语言中有两个指令，即DRAW，XDRAW。前者给出造型，后者消除一个造型，实际上是用互补颜色再给一次造型而达抵消目的。在机器语言中上述两条指令的入口地址不同，前者是＄F601，后者是＄F65D。但调用这两个子程序前，应将旋转值（对应BASIC命令ROT＝N）

放在累加器A中，至于造型表末地址的低位、高位值分别放在X、Y暂存器中（其值为$1A，$1B），即：

```
A6       1A   LDX   $1A
A4       1B   LDY   $1B
A9       00   LDA   #$00
20   5D  F6   JSR   $F65D
```

本例是旋转值为$00，即不旋转。

八、返回

```
60  RTS
```

至此，整个用机器语言调用造型表的设计思想和方法叙述完成。

最后，我们给出一个造型数据见程序GP2A06，它放在$6000—$6072中，在执行程序GP2A05的机器语言子程序前，先将$6000—$6072的造型数据调入内存，然后在BASIC状态下执行CALL768↙，即可看到一辆汽车在屏幕上闪动。

程序GP 2 A06清单。

```
6000-  01  00  05  00  00  41  38  20
6008-  25  2C  2D  65  0C  0C  0C  2D
6010-  2D  2D  2D  2D  15  15  6D  2D
6018-  2D  15  2D  2D  2E  2E  3E  3E
6020-  BF  37  3F  3C  1C  24  1C  3F
6028-  17  17  36  3F  3F  3F  F7  27
6030-  3F  E4  1C  3F  17  3E  3F  2D
6038-  25  6D  15  F6  3F  07  20  64
6040-  2D  04  20  05  28  28  2D  2D
6048-  2D  35  35  35  3F  3F  3F  24
6050-  34  36  3F  3F  3F  4A  49  92
6058-  3A  2D  4E  49  49  20  64  2D
```

```
6060- B6 1E 7F 49 09 2C 25 3C
6068- 2C 2D 2D 3C 3F 3F EF 3F
6070- 3F 27 2D
```

第三节　造型图形的动画显示

利用造型技术，可以把一些实物绘制成造型图，并把它存放在内存中。然后用机器语言子程序或者BASIC程序来调用，就可以形成图形的动画显示。

在这一节中，我们不准备过多地谈及造型原理、造型表存贮、造型绘图命令以及造型表的应用等方面问题（有关这方面内容，请见《中华学习机汉字软件》有关章节），而是给出几个造型图形动画显示的实例。

这些实例不仅富有情趣，而且可以从中学到不少有实用价值的工具软件和组织动画的方法。

一、飞速奔驰的汽车

整个设计由三部分组成：

- 用图形表绘出小汽车。
- 用逐位移动方法编制机器语言。
- 实现汽车动画的BASIC程序控制。

小汽车的图形表放在首地址为$8500，以文件名GP2A07存于磁盘中。造型表见GP2A07。

程序GP2A07清单。

```
8500- 01 00 05 00 00 41 38 20
8508- 25 2C 2D 65 0C 0C 0C 2D
8510- 2D 2D 2D 2D 15 15 AD 2D
8518- 2D 15 2D 2D 2E 2E 3E 3E
```

```
8520- BF 37 3F 3C 1C 24 1C 3F
8528- 17 17 36 3F 3F 3F F7 27
8530- 3F E4 1C 3F 17 3E 3F 2D
8538- 25 6D 15 F6 3F 07 20 64
8540- 2D 04 20 05 28 28 2D 2D
8548- 2D 35 35 35 3F 3F 3F 24
8550- 34 36 3F 3F 3F 4A 49 92
8558- 3A 2D 4E 49 49 20 64 2D
8560- B6 1E 7F 49 09 2C 25 3C
8568- 2C 2D 2D 3C 3F 3F EF 3F
8570- 3F 27 2D 4F 54 52 41 43
```

高分辨率图形逐位平移的机器语言程序，取名为GP2A08，和上面图形表存在同一磁盘中。GP2A08：程序清单。

```
1000- A5 0C 85 05 85 07 A9 50
1008- 85 04 A9 D0 85 06 A2 28
1010- A9 08 85 0F A0 27 B1 04
1018- 2A 2A A0 00 B1 04 2A 91
1020- 04 0A C8 CA D0 F6 A2 28
1028- A0 27 B1 06 2A 2A A0 00
1030- B1 06 2A 91 06 0A C8 CA
1038- D0 F6 A2 28 18 A5 05 69
1040- 04 85 05 85 07 C6 0F D0
1048- CB A5 0C 85 05 85 07 60
```

主程序采用BASIC语言编写，取名为GP2A09，这样运动方法比较简单，RUNGP2A09即可看到小汽车在屏幕上的动态显示。

程序GP2A09清单。

```
10 D$ =  CHR$ (4): POKE 232,00: POKE
      233,133
20  PRINT D$;"BLOAD GP2A10,A$8500"
```

```
30   HGR : HCOLOR= 3
40   ROT= 00: SCALE= 1
50   HOME : POKE 12,33
60   PRINT D$;"BLOAD GP2A08"
70   HPLOT 0,159 TO 279,159
80   FOR I = 1 TO 279
90   CALL 4096
100   DRAW 1 AT I,145
110   NEXT I
120   FOR I = 1 TO 2500: NEXT I
130   TEXT : HOME : END
```

在行进中的小汽车瞬间图形，见图2.68。



图 2.68

几点说明：

① 232（$E8)和233($E9)号单元，是专门用来存放造型表地址的，前者存放造型表首地址的低位（即$8500中的00)，后者存放造型表首地址的高位（即$8500中的85)。只有把造型表送入内存（见BASIC程序20句)，并将首地址装入$E8和$E9（见BASIC程序10句)两个单元中，这个造型表才能使用。

② $8500单元放的是图形定义的个数(本例只有一个造型，所以放01)；$8501单元按造型表存放规则是不用的，所以放00，$8502单元应放的是1号图形定义首地址与$8500的偏移量低位值，即05，而高位值偏移量是00，$8504单元放的也是00，没有什么影响，所以从$8505单元开始才是真

正的小汽车图形数据。而在图形数据放置完毕后，应再用一个单元（本例是＄8573）放00，以示图形资料结束。任何造型的图形造型表必须按上述规则存放。

③ BASIC程序中CALL 4096是执行程序GP2A08，4096是十进制数，它相等于十六进制的＄1000。

④ 100句中DRAW1 AT I，158，是指对第一个造型，在（I，158)位置上画图，注意其中I的变值，因而汽车能动起来。

二、迅速移动的动物

整个程序组织和上面基本类似，只不过造型中的数据和小汽车不一样，它是一架飞机，也可以说像一只小鸟，而在动态显示中则又像是一个骑士骑着骏马飞奔。存贮地址：＄8500—＄8532，文件名为GP2A10。BASIC程序已经稍加改动（见程序GP2A11），不再多述，其中用了程序GP2A08。

造型表GP2A10清单。

```
8500- 01 00 04 00 75 36 2E 2D
8508- 2D 0E 76 76 76 3E 0F 18
8510- 0F 18 0F 18 3F 3F 37 36
8518- 36 36 3F 27 24 24 3F 3F
8520- F7 1E 1E 27 0D 18 2C 20
8528- 0D 18 2C 28 2D 2D 24 6C
8530- 6C 00 1C
```

动态显示瞬间的图象，见图2.69。



图 2.69

应该指出的是，由于机器语言执行速度很快，因而上述

动画设计形态逼真，动态显示效果好，且没有呆滞和闪烁感。

BASIC程序GP2A11清单。

```
10 D$ =  CHR$ (4): POKE 232,00: POKE
     233,133
20  PRINT D$;"BLOADGP2A07,A$8500"
30  HGR : HCOLOR= 3
40  ROT= 00: SCALE= 1
50  HOME : POKE 12,33
60  PRINT D$;"BLOAD GP2A08"
70  HPLOT 0,159 TO 279,159
80  FOR I = 1 TO 279
90  CALL 4096
100  DRAW 1 AT I,158
110  NEXT I
120  FOR I = 1 TO 2500: NEXT I
130  TEXT : HOME : END
```

三、向上移动的机器语言程序

存贮在＄300—＄343单元中的机器语言程序，可以使造型图形向上进行动态显示。

程序GP2A12清单。

```
0300- A9 44 85 E8 A9 03 85 E9
0308- 20 D8 F3 A2 03 20 F0 F6
0310- A9 02 85 E7 A9 AA 85 00
0318- 20 2C 03 20 2C 03 C6 00
0320- A5 00 C9 0A F0 03 4C 18
0328- 03 4C 14 03 A2 01 20 30
0330- F7 A2 64 A0 00 A5 00 20
0338- 11 F4 A9 00 A6 1A A4 1B
0340- 20 5D F6 60
```

程序GP 2A 12有以下几个特点：

① 任何一个造型表（包括索引信息和图形定义），只要放在$0343以后的存贮单元，就可直接运行，显示向上移动的动态图形，而不需要用BASIC程序控制，占用内存少，速度快。

② 可以运行多个造型，适当修改程序可以使造型图形放大，并可改变图形显示的位置。

③ 运行方法简单，在装进机器语言子程序和造型表后，在BASIC状态下执行CALL 768↙，或BRUNGP2A12↙；而在监控状态下键入300G↙即可。

下面给出4个造型表，请特别注意它们都是存贮在从$0344开始的存贮空间中。

① 飞机造型GP2A13

```
0344-  01  00  04  00
0348-  75  36  2E  2D  2D  0E  76  76
0350-  76  3E  0F  18  0F  18  0F  18
0358-  3F  3F  37  36  36  3F  27  24
0360-  24  3F  3F  F7  1E  1E  27  0D
0368-  18  2C  20  0D  18  2C  28  2D
0370-  2D  24  6C  6C  00
```

② 汽车造型GP 2A 14

```
0344-  01  00  05  00
0348-  00  41  38  20  25  2C  2D  65
0350-  0C  0C  0C  2D  2D  2D  2D  2D
0358-  15  15  AD  2D  2D  15  2D  2D
0360-  2E  2E  3E  3E  BF  37  3F  3C
0368-  1C  24  1C  3F  17  17  36  3F
0370-  3F  3F  F7  27  3F  E4  1C  3F
0378-  17  3E  3F  2D  25  6D  15  F6
```

```
0380- 3F 07 20 64 2D 04 20 05
0388- 28 28 2D 2D 2D 35 35 35
0390- 3F 3F 3F 24 34 36 3F 3F
0398- 3F 4A 49 92 3A 2D 4E 49
03A0- 49 20 64 2D B6 1E 7F 49
03A8- 09 2C 25 3C 2C 2D 2D 3C
03B0- 3F 3F EF 3F 3F 27 2D 00
```

③ 手枪造型GP 2A15

```
0344- 01 00 04 00
0348- 2D 2D 2D 2D 2D 2D 35 3F
0350- 3F 3F 3F 3F 3F 37 36 36
0358- 25 24 6C 49 5E 1E 17 35
0360- F8 03 00 3E BF
```

④ 跳伞造型GP 2A16

```
0344- 01 00 04 00
0348- 1A 0D 18 2D 28 0D 76 22
0350- 08 75 75 DF 1E 1E 1E 0F
0358- 18 0F 18 0F 18 4D 08 18
0360- 36 36 D7 0E 6E 2C B0 3A
0368- 3F 3C 2D 2D 3E 3F 6F 13
0370- 3E 4D 39 6C 09 00 1C
```

最后请注意两点：

① 机器语言向上移动的子程序，存贮时不能丢尾巴，即＄0343中的60必须存进去，而且不能是其它值，否则动画显示失败。

② ＄0344单元开始只能存放一个造型，而且＄0344中必须是01，否则显示不起来。

第四节 多个造型图形的动画显示

在许多游戏程序中，常常看到屏幕上有不同图形在移动，引人入胜。这里介绍一个实例，谈谈组织这类程序的方法。

题目是这样的，有两个造型，一是飞机，一是汽车，如何通过一定的手段，让它们在不同时刻向上动画显示，即有时显示飞机，有时显示汽车。

组织这类程序的思路有三点：

・采用搬家的方法，把有关造型搬至某段内存空间。

・按多个造型表设置规则，正确存放造型表。

・修改控制动画显示机器语言子程序有关指针和参数，以便正确调用造型。

一、造型搬家

假设欲把造型表设置在＄6000开始的地址中，并要求把飞机造型和汽车造型同时存贮在＄6000以后的有关单元中。

已知原飞机造型存贮在＄344—＄374中，原汽车造型存贮在＄344—＄3B7中。

先调进飞机造型GP 2A13。

```
0344- 01 00 04 00
0348- 75 36 2E 2D 2D 0E 76 76
0350- 76 3E 0F 18 0F 18 0F 18
0358- 3F 3F 37 36 36 3F 27 24
0360- 24 3F 3F F7 1E 1E 27 0D
0368- 18 2C 20 0D 18 2C 28 2D
0370- 2D 24 6C 6C 00 00
```

其中＄344—＄347是存放的飞机造型的索引信息，真正的图形信息存放在＄348—＄374中，故先应搬家至＄6006 开

始的存贮单元，即：

⁎6006<348·374M↙

逐段扫描 $6006开始的单元，可以看到飞机造型的图形信息已存放在 $6006—$6033中。

然后调进汽车造型GP2A14。

```
0344-  01  00  05  00
0348-  00  41  38  20  25  2C  2D  65
0350-  0C  0C  0C  2D  2D  2D  2D  2D
0358-  15  15  AD  2D  2D  15  2D  2D
0360-  2E  2E  3E  3E  BF  37  3F  3C
0368-  1C  24  1C  3F  17  17  36  3F
0370-  3F  3F  F7  27  3F  E4  1C  3F
0378-  17  3E  3F  2D  25  6D  15  F6
0380-  3F  07  20  64  2D  04  20  05
0388-  28  28  2D  2D  2D  35  35  35
0390-  3F  3F  3F  24  34  36  3F  3F
0398-  3F  4A  49  92  3A  2D  4E  49
03A0-  49  20  64  2D  B6  1E  7F  49
03A8-  09  2C  25  3C  2C  2D  2D  3C
03B0-  3F  3F  EF  3F  3F  27  2D  00
```

同理，$344—$348中存放的是汽车造型的索引信息，原图象信息存放在$349—$3B7中，故再次搬家，移至$6034以后的单元中，即

⁎6034<349·3B7M↙

再次逐段扫描从 $6034开始的单元，发现汽车造型的图形信息已存放在$6034—$60A2中。

至此，两个造型已依次存放完毕，“搬家”任务完成，且程序GP2A18。

二、设置索引

按照存放造型表的规定，在造型图形信息之前，应设置索引信息，安排如下：

$ 6000—02 ；造型个数，本例是 2
$ 6001—00 ；不用，按规定放00
$ 6002—06 ；第一个造型首地址低位
离索引表表头的偏移量
$ 6003—00 ；第一个造型首地址高位
离索引表表头的偏移量
$ 6004—34 ；第二个造型首地址低位
离索引表表头的偏移量
$ 6005—00 ；第二个造型首地址高位
离索引表表头的偏移量

只有按上述规则安排索引表，造型才能被调用。

应该特别指出的是，所有偏移量都是用十六进制数存放的。每个造型数据结束以00为标志。索引表表头$ 6000中的值，应按造型个数实际值安置，否则程序无法执行。

至此，完整的造型表已安排就绪，且程序GP 2A 18。

三、调用造型

调用造型并使造型能够实现动画显示的机器语言子程序，且程序GP 2A 17。

向上移动的机器语言子程序仍用前面介绍的程序GP2A 12。

程序GP 2 A 17基本上就是向上移动的机器语言子程序GP 2A 12，但必须修改某些指针和参数。因为程序GP 2A 12适用于一个造型，而且造型地址在$ 0344中。现在有两个造型，造型表起始地址已放在$ 6000开始的单元，故作以下改

动：

$0301：00

$0305：60　；设置造型表的首地址为$6000

$032D：01　；指一个造型，放02指第二个造型

$0311：01　；不放大，若放大二倍应为02

$0315：80　；指图形起始位置，这里已改为80，原程序为AA

$0332：80　；指图形显示位置，本例改为80

这样，机器语言子程序的指针和参数修改完成，现给出完整的程序GP2A17。

程序GP2A17清单。

```
0300-   A9 00       LDA   #$00
0302-   85 E8       STA   $E8
0304-   A9 60       LDA   #$60
0306-   85 E9       STA   $E9
0308-   20 D8 F3    JSR   $F3D8
030B-   A2 03       LDX   #$03
030D-   20 F0 F6    JSR   $F6F0
0310-   A9 01       LDA   #$01
0312-   85 E7       STA   $E7
0314-   A9 80       LDA   #$80
0316-   85 00       STA   $00
0318-   20 2C 03    JSR   $032C
031B-   20 2C 03    JSR   $032C
031E-   C6 00       DEC   $00
0320-   A5 00       LDA   $00
0322-   C9 0A       CMP   #$0A
0324-   F0 03       BEQ   $0329
0326-   4C 18 03    JMP   $0318
0329-   4C 14 03    JMP   $0314
032C-   A2 02       LDX   #$02
032E-   20 30 F7    JSR   $F730
```

```
0331-   A2 80        LDX   #$80
0333-   A0 00        LDY   #$00
0335-   A5 00        LDA   $00
0337-   20 11 F4     JSR   $F411
033A-   A9 00        LDA   #$00
033C-   A6 1A        LDX   $1A
033E-   A4 1B        LDY   $1B
0340-   20 5D F6     JSR   $F65D
0343-   60           RTS
```

程序 GP2A18清单。

```
6000- 02 00 06 00 34 00 75 36
6008- 2E 2D 2D 0E 76 76 76 3E
6010- 0F 18 0F 18 0F 18 3F 3F
6018- 37 36 36 36 3F 27 24 24
6020- 3F 3F F7 1E 1E 27 0D 18
6028- 2C 20 0D 18 2C 28 2D 2D
6030- 24 6C 6C 00 41 38 20 25
6038- 2C 2D 65 0C 0C 0C 2D 2D
6040- 2D 2D 2D 15 15 AD 2D 2D
6048- 15 2D 2D 2E 2E 3E 3E BF
6050- 37 3F 3C 1C 24 1C 3F 17
6058- 17 36 3F 3F 3F F7 27 3F
6060- E4 1C 3F 17 3E 3F 2D 25
6068- 6D 15 F6 3F 07 20 64 2D
6070- 04 20 05 28 28 2D 2D 2D
6078- 35 35 35 3E 3F 3F 24 34
6080- 36 3F 3F 3F 4A 49 92 3A
6088- 2D 4E 49 49 20 64 2D B6
6090- 1E 7F 49 09 2C 25 3C 2C
6098- 2D 2D 3C 3F 3F EF 3F 3F
60A0- 27 2D 00 14 88 91 14 C8
```

运行方法是：

]BLOADGP2A17↙

・]BLOADGP 2 A 18↙

若运行飞机动画：

]POKE 813，1↙

]CALL 768↙

若运行汽车动画：

]POKE 813，2↙

]CALL 768↙

第五节　动画显示的键盘操作控制

在进行游戏或显示动画程序时，操作者可以通过一定的手段，例如用键盘、游戏杆对它们实行控制，进行诸如上、下、左、右移动；加快、减慢、前进、后退操作；旋转、滚动、联动、变形演示；以及暂停、继续、清除、退出处理。

这里仅对键盘操作控制，介绍几个实例：

一、对画面进行暂停、继续、退出操作控制

程序共分三个部分：

・$300—$32B　；用机器语言调用造型表

・$32C—$350　；实行暂停、继续、退出控制

・$6000—$6030　；一个小鸟画面的造型表

前二部分取名GP 2 A 19，存盘备用，后一部分取名GP 2 A 20存盘备用。

运行方法很简单：

] BLOADGP 2A20，A $6000↙

] BLOADGP 2A19，A $300↙

] CALL 812↙　或✻32CG↙

屏幕上显示一个不断显示不断清除的小鸟画面。按SP-

ACE（空格键），画面暂停不再闪动；按RETURN。（回车键），画面继续运动；按ESC（ESCAPE）键退出。

程序GP2A19清单。

```
0300-   20 E2 F3    JSR   $F3E2
0303-   A9 00       LDA   #$00
0305-   85 E8       STA   $E8
0307-   A9 60       LDA   #$60
0309-   85 E9       STA   $E9
030B-   A2 03       LDX   #$03
030D-   20 F0 F6    JBR   $F6F0
0310-   A9 03       LDA   #$03
0312-   85 E7       STA   $E7
0314-   A2 01       LDX   #$01
0316-   20 30 F7    JSR   $F730
0319-   A2 80       LDX   #$80
031B-   A0 00       LDY   #$00
031D-   A9 32       LDA   #$32
031F-   20 11 F4    JSR   $F411
0322-   A6 1A       LDX   $1A
0324-   A4 1B       LDY   $1B
0326-   A9 00       LDA   #$00
0328-   20 5D F6    JSR   $F65D
032B-   60          RTS
032C-   AD 00 C0    LDA   $C000
032F-   C9 9B       CMP   #$9B
0331-   D0 01       BNE   $0334
0333-   60          RTS
0334-   C9 8D       CMP   #$8D
0336-   F0 0E       BEQ   $0346
0338-   C9 A0       CMP   #$A0
033A-   D0 0A       BNE   $0346
033C-   A9 00       LDA   #$00
033E-   8D 10 C0    STA   $C010
```

```
0341-    AD 00 C0    LDA   $C000
0344-    10 FB       BPL   $0341
0346-    A9 00       LDA   #$00
0348-    8D 10 C0    STA   $C010
034B-    20 00 03    JSR   $0300
034E-    4C 2C 03    JMP   $032C
```

程序GP2A20清单。

```
6000- 01 00 04 00 75 36 2E 2D
6008- 2D 0E 76 76 76 3E 0F 18
6010- 0F 18 0F 18 3F 3F 37 36
6018- 36 36 3F 27 24 24 3F 3F
6020- F7 1E 1E 27 0D 18 2C 20
6028- 0D 18 2C 28 2D 2D 24 6C
6030- 6C 00
```

这里将$32C—$350程序段作一解释：首先读键盘，判断是否是ESC键（键盘字符的ASCII码是$9B），是的退出，不是的和RETURN的代码$8D比较，是的继续程序运行，不是的再和SPACE比较（代码是$AO），是的暂停，不是的继续。其中$C000单元为键盘数据锁存器的地址，而$C010单元则是清除键盘数据锁存器选通信号开关的地址。这就是说用相应命令可以完成将键盘数据锁存器的值读入累加器，和清除键盘选通信号。

二、对不同动画图形进行选择的操作控制

例如，有两个动画造型放在$6000—$60A2单元中，其中$6000—$6005为这两个动画的索引信息，$6006—$6033放的是飞机造型数据，$6034—$60A2放的是汽车造型(见本章第四节)。为了任意选择其中一个造型进行动态显示，必须安排适当的控制键。

调用造型并显示动画的机器语言子程序放在＄300—＄343中。

键盘操作控制放在＄344—＄37D中。

这段操作要求除按ESC键退出，按RETURN键继续，按SPACE键暂停外，还要求加两个控制键1和2。按1选择飞机动画，按2选择汽车动画。

两个造型的造型表，存在＄6000—＄60A2中，即前面介绍过的程序GP2A18。

调用造型并对其实现控制的程序，放在＄300—＄37D中，程序名为GP2A21。

```
0300- A9 00 85 E8 A9 60 85 E9
0308- 20 D8 F3 A2 03 20 F0 F6
0310- A9 01 85 E7 A9 80 85 00
0318- 20 2C 03 20 2C 03 C6 00
0320- A5 00 C9 0A F0 03 4C 18
0328- 03 4C 44 03 A2 01 20 30
0330- F7 A2 80 A0 00 A5 00 20
0338- 11 F4 A9 00 A6 1A A4 1B
0340- 20 5D F6 60 AD 00 C0 C9
0348- 9B D0 01 60 C9 B1 F0 16
0350- C9 B2 F0 1A C9 8D F0 1B
0358- C9 A0 D0 17 A9 00 8D 10
0360- C0 AD 00 C0 10 FB A9 01
0368- 8D 2D 03 4C 73 03 A9 02
0370- 8D 2D 03 A9 00 8D 10 C0
0378- 20 00 03 4C 44 03
```

键盘操作控制对应的汇编语言见程序GP2A22。

程序GP2A22清单。

```
0344-    AD 00 C0     LDA    $C000
0347-    C9 9B        CMP    #$9B
0349-    D0 01        BNE    $034C
```

```
034B-   60         RTS
034C-   C9 B1      CMP   #$B1
034E-   F0 16      BEQ   $0366
0350-   C9 B2      CMP   #$B2
0352-   F0 1A      BEQ   $036E
0354-   C9 8D      CMP   #$8D
0356-   F0 1B      BEQ   $0373
0358-   C9 A0      CMP   #$A0
035A-   D0 17      BNE   $0373
035C-   A9 00      LDA   #$00
035E-   8D 10 C0   STA   $C010
0361-   AD 00 C0   LDA   $C000
0364-   10 FB      BPL   $0361
0366-   A9 01      LDA   #$01
0368-   8D 2D 03   STA   $032D
036B-   4C 73 03   JMP   $0373
036E-   A9 02      LDA   #$02
0370-   8D 2D 03   STA   $032D
0373-   A9 00      LDA   #$00
0375-   8D 10 C0   STA   $C010
0378-   20 00 03   JSR   $0300
037B-   4C 44 03   JMP   $0344
```

键盘操作控制的思路同前面介绍的第一个键盘操作控制，但因增加了对两个造型的选择控制，故需增加相应的程序段，并修改必要的地址。

在＄32D单元中存放的值，是决定显示第几个造型的，放01选中第一个造型，放02选中第二个造型。因此，为了实现控制，键盘操作控制段应增加对那一个造型的判断。键盘上“1”的ASCII码是B 1,“2”的ASCII码是B 2。

现在对＄344—＄37D内容做一解释。

首先读键盘，判断是否是ESC键，是的结束，不是的判断是否是“1”,是的转 ＄366,取“01”放在＄032D中，再

转＄0373清键盘，继而转＄300开始的动画显示程序，选中第一个造型飞机。若不是“1”，再判断是不是“2”，是的转＄036E，取“02”放在＄032D中，再清键盘，继续选中第二个汽车造型。不是“2”，再判断是否是RETURN键、SPC键等，过程同前分析。

三、对画面进行上移、下移、暂停、结束操作控制

机器语言子程序GP 2A 23可以实现对画面上移(按U键，对应的ASCII 码是$DS)、下移（按D键，对应的ASCII码是$C 4）、暂停（按SPACE 键，对应的ASCII码是$AO)等操作的。其对应的汇编语言存贮在＄1046—＄1097单元中；原理读者可以自行阅读。

程序GP 2A 23清单。

```
1046-   AD 00 C0     LDA   $C000
1049-   10 C1        BPL   $100C
104B-   C9 A0        CMP   #$A0
104D-   D0 08        BNE   $1057
104F-   2C 10 C0     BIT   $C010
1052-   AD 00 C0     LDA   $C000
1055-   10 FB        BPL   $1052
1057-   C9 8D        CMP   #$8D
1059-   F0 36        BEQ   $1091
105B-   C9 D5        CMP   #$D5
105D-   D0 17        BNE   $1076
105F-   A9 00        LDA   #$00
1061-   8D 0D 10     STA   $100D
1064-   A9 E8        LDA   #$E8
1066-   8D 18 10     STA   $1018
1069-   A9 01        LDA   #$01
106B-   8D 24 10     STA   $1024
106E-   A9 C0        LDA   #$C0
1070-   8D 39 10     STA   $1039
```

```
1073-   4C 09 10    JMP   $1009
1076-   C9 C4       CMP   #$C4
1078-   D0 8F       BNE   $1009
107A-   A9 BF       LDA   #$BF
107C-   8D 0D 10    STA   $100D
107F-   A9 CA       LDA   #$CA
1081-   8D 18 10    STA   $1018
1084-   A9 BE       LDA   #$BE
1086-   8D 24 10    STA   $1024
1089-   A9 FF       LDA   #$FF
108B-   8D 39 10    STA   $1039
108E-   4C 09 10    JMP   $1009
1091-   2C 10 C0    BIT   $C010
1094-   4C 69 FF    JMP   $FF69
1097-   60          RTS
```

实例：图形和文字的上下滚动显示。

这里提供一个图形和文字上下滚动显示的程序，其效果像电影开始时演员表一样。在编制软件开始时，运行一幅精美的图象，或者显示一个移动式的中文菜单，以及介绍程序的设计思想、功能和操作方法等，都很方便和实用。

本程序由两个机器语言子程序（UPDOWN和UPD）及一个BASIC程序（UPD-1）组成。建立和实施本程序的方法如下：

首先，在监控状态下，键入300-31D一段存贮单元的机器语言子程序，并将其取名为UPDOWN，存盘备用。

然后，再在监控状态下，键入1000—1097一段存贮单元的机器语言子程序，取名为UPD存盘备用。

最后，键入文件名为UPD-1的BASIC程序，同样存盘备用。

画一幅图形（例如PIC 1)，将其存在高分辨图形第一页

面。

程序 UPDOWN 清单。

```
0300- A9 C0 A0 00 A2 00 20 11
0308- F4 A0 C0 A5 26 91 06 A5
0310- 27 91 08 EE 01 03 EE 0A
0318- 03 C0 BF 90 E3 60
```

程序 UPD 清单。

```
1000- AD 50 C0 AD 57 C0 AD 52
1008- C0 AD 10 C0 A2 BF BD 00
1010- 0C 85 FE BD C0 0C 85 FF
1018- CA BD 00 0C 85 FC BD C0
1020- 0C 85 FD E0 BE D0 08 A0
1028- 27 B1 FE 48 88 10 FA A0
1030- 27 B1 FC 91 FE 88 10 F9
1038- E0 FF D0 D2 A0 00 68 91
1040- FE C8 C0 28 90 F8 AD 00
1048- C0 10 C1 C9 A0 D0 08 2C
1050- 10 C0 AD 00 C0 10 FB C9
1058- 8D F0 36 C9 D5 D0 17 A9
1060- 00 8D 0D 10 A9 E8 8D 18
1068- 10 A9 01 8D 24 10 A9 C0
1070- 8D 39 10 4C 09 10 C9 C4
1078- D0 8F A9 BF 8D 0D 10 A9
1080- CA 8D 18 10 A9 BE 8D 24
1088- 10 A9 FF 8D 39 10 4C 09
1090- 10 2C 10 C0 4C 69 FF 60
```

程序 UPD -1清单。

```
10 D$ =  CHR$ (4)
20  PRINT D$"BLOADPIC 1,A$2000"
30  PRINT D$"BLOADUPDOWN"
40  PRINT D$"BLOADUPD"
50  POKE 6,0: POKE 7,12: POKE 8,19
     2: POKE 9,12
60  POKE 230,32
```

```
70  CALL 768
80  CALL 4096
```

操作方法：

只要直接运行UPD-1程序即可。程序运行后可以看到PIC 1的图形向上滚动显示。而在翻滚的过程中，按D键，图形改变方向，向下滚动显示；按U键，图形又变为向上滚动显示；若要暂定，可按一下空格键；继续进行则按D或U键；结束运行按CTRL-RESET后返回BASIC。

将程序UPD-1稍加改动，便可运行第一页和第二页的图形，见程序UPD-2。运行方法是首先RUN UPD-2，按CTRL-RESET后，返回BASIC，此时执行RUN90↙（或GOTO 90↙），则立即显示滚动的第二页面。在运行过程中，D、U、空格键和CTRL-RESET的功能同上。

程序UPD-2及瞬间显示的文字图形：

```
10 D$ =  CHR$ (4)
20  PRINT D$"BLOADPIC 1,A$2000"
25  PRINT D$"BLOADAAAAA,A$4000"
30  PRINT D$"BLOADUPDOWN"
40  PRINT D$"BLOADUPD"
50  POKE 6,0: POKE 7,12: POKE 8,19
     2: POKE 9,12
60  POKE 230,32
70  CALL 768
80  CALL 4096
90 Y$ = "2000<4000.5FFFM N D823G"
100  FOR I = 1 TO  LEN (Y$): POKE
     511 + I, ASC ( MID$ (Y$,I,1))
      + 128: NEXT I: POKE 72,0
110  CALL  - 144
```

```
120  CALL 768
130  CALL 4096
```

```
            全屏幕绘图软件
Y—左上                        G—左下
B—右下                        H—右上
I—向上                        M—向下
K—向右                        J—向左
L—画图                        O—擦图
C—清除                        T—保留
S—存盘                        A—调盘
W—打印                        E—结束
            按空格键运行
```

第六节　造型动画设计实例

机器语言程序繁琐而不直观，使用它来绘图需要下不少功夫。但它有高级语言所没有的许多特点，其中之一就是动画设计。由于机器语言执行速度很快，因而动画效果较好。

下面介绍一个简单的动画设计程序。运行这个程序，屏幕进入高分辨作图状态，一个小动物从屏幕左端到右端跳跃前进，同时在其尾部画出运动的轨迹。不断重复这一跳跃过程，运动轨迹就在一定的幅度内组成彩带状图形。

小动物是一个造型图，存放在机器语言程序后＄8CF开始的地址中，见程序GP 2A24。

```
08CF- 01
08D0- 00 04 00 36 F6 1E D7 39
08D8- 3F 0F 18 0F 18 0F 18 24
08E0- 2C 28 28 28 2D 15 0E DF
08E8- DB DB 4B 49 89 92 09 A8
```

```
08F0- F3 3F 20 0D 18 25 08 18
08F8- 08 D8 DB 9B 49 23 0F 18
0900- 0F 18 37 35 35 DF DB 0B
0908- 18 08 58 49 96 92 08 18
0910- 18 08 D8 0E 0E 0E 0E F4
0918- 0F 18 0F 18 0F 18 4D 49
0920- DA B3 92 52 49 49 49 1A
0928- 96 08 18 16 18 F8 DB DB
0930- DB 08 18 08 18 08 18 48
0938- 91 0A 18 3F 38 0F 18 3F
0940- 3F F7 F7 1E 96 12 08 18
0948- 08 D8 70 D9 71 73 0E 0E
0950- 0E 2D 2D 2D 25 08 18 90
0958- FD 3F DF DB 52 49 49 49
0960- F8 DB DB 08 18 08 18 0E
0968- 70 F6 0B 1E 4D 49 49 49
0970- D9 0A 3F 2C 3C 3C 6F 21
0978- 08 18 08 18 08 18 08 D8
0980- DB DB DB DB 00
```

其中＄8CF中存放造型数目(＄01个)，＄8D1、＄8D2为造型始址的偏移量（＄0004)，实际该造型存放在＄8D3—＄984的内存单元中。

程序部分通过对这一造型的显示、清除和移位来达到动画的效果，见程序GP 2A 25。

```
0801-    A9 CF         LDA    #$CF
0803-    85 E8         STA    $E8
0805-    A9 08         LDA    #$08
0807-    85 E9         STA    $E9
0809-    20 D8 F3      JSR    $F3D8
080C-    A2 03         LDX    #$03
080E-    20 F0 F6      JSR    $F6F0
0811-    A9 01         LDA    #$01
0813-    85 E7         STA    $E7
0815-    A9 64         LDA    #$64
```

```
0817-   85 00       STA   $00
0819-   A9 19       LDA   #$19
081B-   85 01       STA   $01
081D-   A9 01       LDA   #$01
081F-   85 02       STA   $02
0821-   A2 01       LDX   #$01
0823-   20 30 F7    JSR   $F730
0826-   20 90 08    JSR   $0890
0829-   20 BD 08    JSR   $08BD
082C-   A5 00       LDA   $00
082E-   C9 64       CMP   #$64
0830-   D0 13       BNE   $0845
0832-   A5 01       LDA   $01
0834-   18          CLC
0835-   65 02       ADC   $02
0837-   38          SEC
0838-   E9 19       SBC   #$19
083A-   AA          TAX
083B-   A0 00       LDY   #$00
083D-   A9 66       LDA   #$66
083F-   38          SEC
0840-   E5 02       SBC   $02
0842-   20 57 F4    JSR   $F457
0845-   E6 02       INC   $02
0847-   A5 02       LDA   $02
0849-   C9 15       CMP   #$15
084B-   F0 03       BEQ   $0850
084D-   4C 21 08    JMP   $0821
```

20 × 20 = 400

```
0850-   A5 01       LDA   $01
0852-   18          CLC
0853-   69 14       ADC   #$14
0855-   85 01       STA   $01
0857-   A9 01       LDA   #$01
0859-   85 02       STA   $02
085B-   20 C6 08    JSR   $08C6
085E-   20 C6 08    JSR   $08C6
```

```
0861-    A5 00        LDA   $00
0863-    C9 64        CMP   #$64
0865-    D0 13        BNE   $087A
0867-    A5 01        LDA   $01
0869-    18           CLC
086A-    65 02        ADC   $02
086C-    38           SEC
086D-    E9 19        SBC   #$19
086F-    AA           TAX
0870-    A0 00        LDY   #$00
0872-    A9 52        LDA   #$52
0874-    18           CLC
0875-    65 02        ADC   $02
0877-    20 57 F4     JSR   $F457
087A-    E6 02        INC   $02
087C-    A5 02        LDA   $02
087E-    C9 15        CMP   #$15
0880-    F0 03        BEQ   $0885
0882-    4C 5B 08     JMP   $085B
0885-    A5 01        LDA   $01
0887-    69 14        ADC   #$14
0889-    85 01        STA   $01
088B-    E6 01        INC   $01
088D-    4C 1D 08     JMP   $081D
0890-    A5 01        LDA   $01
0892-    18           CLC
0893-    65 02        ADC   $02
0895-    AA           TAX
0896-    A0 00        LDY   #$00
0898-    A5 00        LDA   $00
089A-    38           SEC
089B-    E5 02        SBC   $02
089D-    20 11 F4     JSR   $F411
08A0-    A9 00        LDA   #$00
08A2-    A6 1A        LDX   $1A
08A4-    A4 1B        LDY   $1B
08A6-    20 5D F6     JSR   $F65D
```

```
08A9-    60            RTS
08AA-    A5 01         LDA   $01
08AC-    18            CLC
08AD-    65 02         ADC   $02
08AF-    AA            TAX
08B0-    A0 00         LDY   #$00
08B2-    A5 00         LDA   $00
08B4-    38            SEC
08B5-    E9 14         SBC   #$14
08B7-    18            CLC
08B8-    65 02         ADC   $02
08BA-    4C 9D 08      JMP   $089D
08BD-    A2 01         LDX   #$01
08BF-    20 30 F7      JSR   $F730
08C2-    20 90 08      JSR   $0890
08C5-    60            RTS
08C6-    A2 01         LDX   #$01
08C8-    20 30 F7      JSR   $F730
08CB-    20 AA 08      JSR   $08AA
08CE-    60            RTS
```

程序GP 2 A 25注释：

$0801—$0808；设置造型表始址$08CF。

$0809—$0810；启动第二高分辨作图页，设置颜色为白（3）。

$0811—$0814；造型放大倍数为1（不放大）。

$0815—$0820；参量初始化。

$0821—$082B；显示或清除造型。

$082C—$0844；画出尾部轨迹。

$0845—$084F；改变纵坐标。

$0850—$0860；显示或清除造型。

$0861—$0879；画出尾部轨迹。

$ 087A—$ 0884：改变纵坐标。

$ 0885—$ 088F：改变横坐标。

$ 0890—$ 08A 9：显示或清上跃中的造型。

$ 08AA—$ 08BC：显示或清除下落中的造型。

首先，将造型表起始地址的低、高位分别存入零页指针单元$ E 8（232）、$ E 9（233）中，启动高分辨作图第二页并设置颜色为白色（3）。程序中的20D8F3指令即相当于HGR 2，而A 2 03和20F0F 6两条指令相当于HCOLOR＝3

约定造型显示时不作放大处理，即将放大倍数1存入零页单元$ E 7。从$ 811—$ 820因参量初值化程序，它设定动画的起始位置为横坐标$ 19(25)、纵坐标$ 64(100)，跳跃

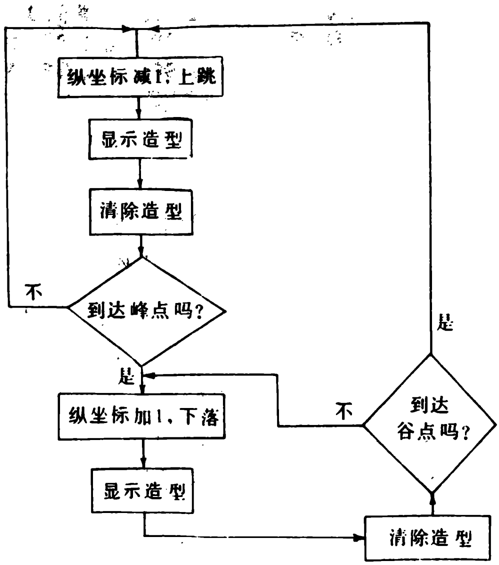


图 2.70

幅度起始值为＄01(1)。

＄821开始正式进入动画设计，其流程大致见图2.70。

即分上跳和下落两个过程，上跳时纵坐标递增，下落时纵坐标递增。其中最主要的功能就是造型的显示（或清除)。

造型的显示通过调用几个系统子程序来实现：

① 确定显示的造型序号

这一序号就放在暂存器X中，然后调用＄F730 的子程序，即：

```
A2 01 LDX #$01
20 30 F7 JSR $F730
```

② 确定显示位置

调用子程序前先将纵坐标存入累加器A，横坐标的高、低位元分别存入暂存器Y和X中，然后转＄F411地址执行。

③ 显示造型

显示造型的子程序在＄F601和＄F65D中(分别相当于DRAW或XDRAW指令)，调用前先将旋转值存入累加器A，＄1A、＄1B中的地址指针存入暂存器X和Y中，即

```
A9  00     LDA #$00
A6  1A     LDX $1A
A4  1B     LDY $1B
20  5DF6   JSR $F65D
```

以上三个步骤合起来相当于语句：

XDRAW A AT X，Y

其中A为造型序号，X和Y为横坐标和纵坐标。

将程序GP 2A 25输入＄801开始的内存中，使用801 G或CALL 2049指令即可执行。

第十一章　多页联动

使文字或图象在屏幕上移动，是编制教学辅助软件或游戏程序常用的一种技巧。而在高分辨图形显示的一个页面上，实现两页、三页以至更多页图形或文字的联动显示，将使画面更加丰富多采。

本章主要介绍二页、三页图象的联动显示技术，至于四页、五页和更多页图象的联动显示，我们将放在特殊技巧一章中介绍。

第一节　两页图象上下联动显示

我们曾在本篇第六章移动显示中，介绍了单页图象的移动显示，实际上两页图象的联动显示，可以在单页图象移动显示程序的基础上，稍作一些改动，即可实现。

程序GP2B 01为两页图象向上移动显示的机器语言子程序，其原理见图2.71。

程序GP2B 01实际上是将GP2603、GP2605这两个程序结合起来编制的。其原理说明如下：

$8000—$801A　　　同程序GP 2605，取得地址。

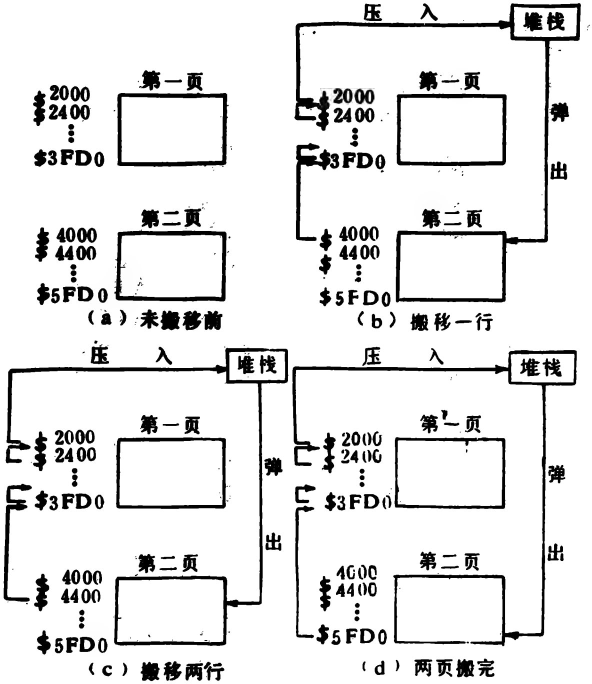


图 2.71

程序GB2B01清单。

```
8000-   A9 00       LDA   #$00
8002-   85 1F       STA   $1F
8004-   A2 00       LDX   #$00
8006-   BD 00 81    LDA   $8100,X
8009-   85 07       STA   $07
800B-   BD 00 82    LDA   $8200,X
800E-   85 06       STA   $06
8010-   E8          INX
8011-   BD 00 81    LDA   $8100,X
```

```
8014-   85 09       STA   $09
8016-   BD 00 82    LDA   $8200,X
8019-   85 08       STA   $08
801B-   E0 01       CPX   #$01
801D-   D0 08       BNE   $8027
801F-   A0 27       LDY   #$27
8021-   B1 06       LDA   ($06),Y
8023-   48          PHA
8024-   88          DEY
8025-   10 FA       BPL   $8021
8027-   A0 27       LDY   #$27
8029-   B1 08       LDA   ($08),Y
802B-   91 06       STA   ($06),Y
802D-   88          DEY
802E-   10 F9       BPL   $8029
8030-   E0 BF       CPX   #$BF
8032-   D0 D2       BNE   $8006
8034-   A5 08       LDA   $08
8036-   85 06       STA   $06
8038-   A5 09       LDA   $09
803A-   85 07       STA   $07
803C-   A6 1F       LDX   $1F
803E-   BD 00 81    LDA   $8100,X
8041-   18          CLC
8042-   69 20       ADC   #$20
8044-   85 09       STA   $09
8046-   BD 00 82    LDA   $8200,X
8049-   85 08       STA   $08
804B-   A0 27       LDY   #$27
804D-   B1 08       LDA   ($08),Y
804F-   91 06       STA   ($06),Y
8051-   88          DEY
8052-   10 F9       BPL   $804D
8054-   A0 00       LDY   #$00
8056-   68          PLA
8057-   91 08       STA   ($08),Y
8059-   C8          INY
```

```
805A-   C0 28      CPY   #$28
805C-   90 F8      BCC   $8056
805E-   E6 1F      INC   $1F
8060-   A5 1F      LDA   $1F
8062-   C9 C0      CMP   #$C0
8064-   D0 9E      BNE   $8004
8066-   60         RTS
8067-   A9 20      LDA   #$20
8069-   85 E6      STA   $E6
806B-   A2 00      LDX   #$00
806D-   8A         TXA
806E-   48         PHA
806F-   20 11 F4   JSR   $F411
8072-   68         PLA
8073-   AA         TAX
8074-   A5 26      LDA   $26
8076-   9D 00 82   STA   $8200,X
8079-   A5 27      LDA   $27
807B-   9D 00 81   STA   $8100,X
807E-   E8         INX
807F-   E0 C0      CPX   #$C0
8081-   D0 EA      BNE   $806D
8083-   AD 54 C0   LDA   $C054
8086-   AD 50 C0   LDA   $C050
8089-   AD 57 C0   LDA   $C057
808C-   AD 52 C0   LDA   $C052
808F-   AD 10 C0   LDA   $C010
8092-   20 00 80   JSR   $8000
8095-   AD 00 C0   LDA   $C000
8098-   10 F5      BPL   $808F
809A-   60         RTS
```

$801B－$8033　同子程序GP 2603，首行压栈，其余各行上移一行，判断移完否，未完继续移。

$8034—$804A	移完，将第一页末行地址改为目的行地址，并取第二页被移行地址。
$804B—$8053	移动一行。
$8054—$805D	堆栈弹出，见图2.71（b)。
$805E—$8065	计数器加1，判断是否全部移完，未完继续移见图2.75（c)，(d)。
$8066	移完返回。
$8067—$809A	演示用子程序，同GP 2603。

演示子程序GP 2B01，可用程序GP 2B02。

```
10 D$ =  CHR$ (4)
20  PRINT D$"BLOAD PIC-1,A$2000"
30  PRINT D$"BLOAD PIC-2,A$4000"
40  PRINT D$"BLOAD GP2B01"
50  CALL 32871
60  FOR I = 1 TO 1000: NEXT I
70  TEXT : HOME : END
```

欲使两页图形向下移动，可将程序GB2B01中$8001单元内容改为$BF；$8004单位内容也改为$BF；$8010单元内容改为$CA；$8010单元内容改为$BE；$8031单元内容改为$00；$8063单元内容改为$FF即可，见程序GP2B03。原理不再重复。

第二节　三页图象上下联动显示

两页图象上下联动显示的原理搞清楚后，要实现三页图象上下联动显示也就很容易了。三页图象上移显示原理见图2.72。将两页图象移动显示子程序改动一下，在两页移完后不结束而继续将第三页图象移往第一页显示，并将第一页压栈的内容弹出至第三页，循环往复直至第三页图象移完，见程

序GP2B 04。

其原理简述如下：

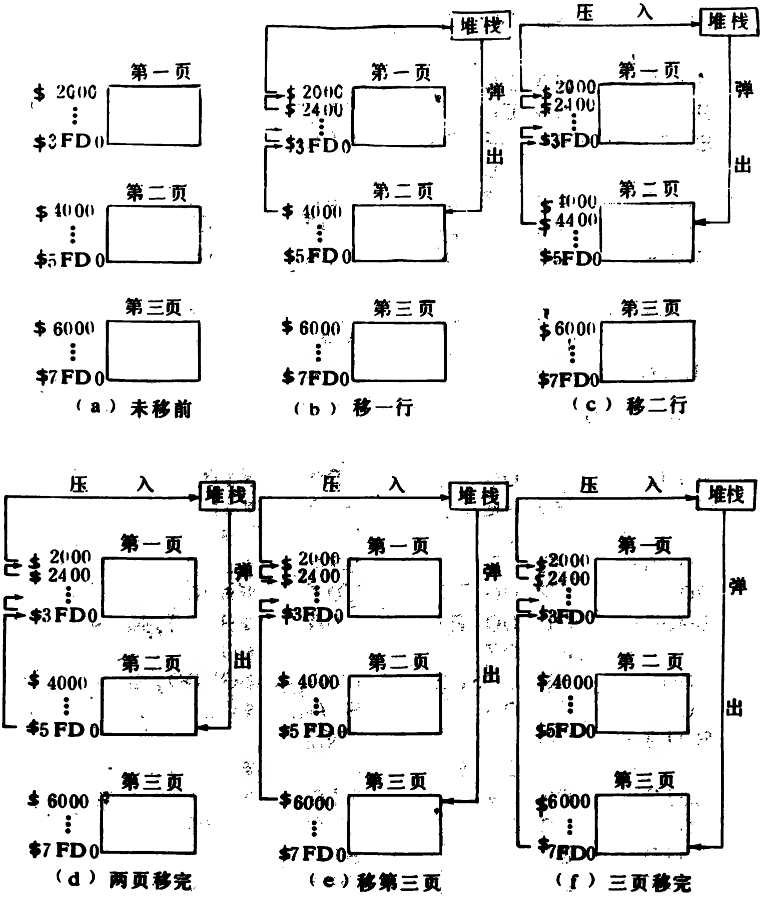
$8000—$8003　　增设一页数计数器，用以记录图形页数，$1E内容为0时图象为第一页，为1时图象为第二页，为2时图象为第三页。

$8004—$8037　　同子程序GP2B 01，将第一页图象首行压栈，其余各行全部上移一行，并判断移完否。

程序GP2B03清单。

```
8000-   A9 BF       LDA   #$BF
8002-   85 1F       STA   $1F
8004-   A2 BF       LDX   #$BF
8006-   BD 00 81    LDA   $8100,X
8009-   85 07       STA   $07
800B-   BD 00 82    LDA   $8200,X
800E-   85 06       STA   $06
8010-   CA          DEX
8011-   BD 00 81    LDA   $8100,X
8014-   85 09       STA   $09
8016-   BD 00 82    LDA   $8200,X
8019-   85 08       STA   $08
801B-   E0 BE       CPX   #$BE
801D-   D0 08       BNE   $8027
801F-   A0 27       LDY   #$27
8021-   B1 06       LDA   ($06),Y
8023-   48          PHA
8024-   88          DEY
8025-   10 FA       BPL   $8021
8027-   A0 27       LDY   #$27
8029-   B1 08       LDA   ($08),Y
802B-   91 06       STA   ($06),Y
802D-   88          DEY
```

```
802E-   10 F9       BPL   $8029
8030-   E0 00       CPX   #$00
8032-   D0 D2       BNE   $8006
8034-   A5 08       LDA   $08
8036-   85 06       STA   $06
8038-   A5 09       LDA   $09
803A-   85 07       STA   $07
803C-   A6 1F       LDX   $1F
803E-   BD 00 81    LDA   $8100,X
8041-   18          CLC
8042-   69 20       ADC   #$20
8044-   85 09       STA   $09
8046-   BD 00 82    LDA   $8200,X
8049-   85 08       STA   $08
804B-   A0 27       LDY   #$27
804D-   B1 08       LDA   ($08),Y
804F-   91 06       STA   ($06),Y
8051-   88          DEY
8052-   10 F9       BPL   $804D
8054-   A0 00       LDY   #$00
8056-   68          PLA
8057-   91 08       STA   ($08),Y
8059-   C8          INY
805A-   C0 28       CPY   #$28
805C-   90 F8       BCC   $8056
805E-   C6 1F       DEC   $1F
8060-   A5 1F       LDA   $1F
8062-   C9 FF       CMP   #$FF
8064-   D0 9E       BNE   $8004
8066-   60          RTS
8067-   A9 20       LDA   #$20
8069-   85 E6       STA   $E6
806B-   A2 00       LDX   #$00
806D-   8A          TXA
806E-   48          PHA
806F-   20 11 F4    JSR   $F411
8072-   68          PLA
```



第一页
$ 2000
$3FD0
第二页
$ 4000
$5FD0
第三页
$ 6000
$7FD0
（a）未移前
堆栈
弹
出
$ 2400
（b）移一行
压 入
$ 4400
（c）移二行
（d）两页移完
（e）移第三页
（f）三页移完

图 2.72

```
8073-   AA          TAX
8074-   A5 26       LDA   $26
8076-   9D 00 82    STA   $8200,X
8079-   A5 27       LDA   $27
807B-   9D 00 81    STA   $8100,X
807E-   E8          INX
807F-   E0 C0       CPX   #$C0
8081-   D0 EA       BNE   $806D
8083-   AD 54 C0    LDA   $C054
8086-   AD 50 C0    LDA   $C050
8089-   AD 57 C0    LDA   $C057
808C-   AD 52 C0    LDA   $C052
808F-   AD 10 C0    LDA   $C010
8092-   20 00 80    JSR   $8000
8095-   AD 00 C0    LDA   $C000
8098-   10 F5       BPL   $808F
809A-   60          RTS
```

程序GP2B04清单

```
8000-   A9 00       LDA   #$00
8002-   85 1E       STA   $1E
8004-   A9 00       LDA   #$00
8006-   85 1F       STA   $1F
8008-   A2 00       LDX   #$00
800A-   BD 00 81    LDA   $8100,X
800D-   85 07       STA   $07
800F-   BD 00 82    LDA   $8200,X
8012-   85 06       STA   $06
8014-   E8          INX
8015-   BD 00 81    LDA   $8100,X
8018-   85 09       STA   $09
801A-   BD 00 82    LDA   $8200,X
801D-   85 08       STA   $08
801F-   E0 01       CPX   #$01
8021-   D0 08       BNE   $802B
8023-   A0 27       LDY   #$27
8025-   B1 06       LDA   ($06),Y
```

```
8027-   48          PHA
8028-   88          DEY
8029-   10 FA       BPL   $8025
802B-   A0 27       LDY   #$27
802D-   B1 08       LDA   ($08),Y
802F-   91 06       STA   ($06),Y
8031-   88          DEY
8032-   10 F9       BPL   $802D
8034-   E0 BF       CPX   #$BF
8036-   D0 D2       BNE   $800A
8038-   A5 1E       LDA   $1E
803A-   C9 00       CMP   #$00
803C-   D0 08       BNE   $8046
803E-   A9 20       LDA   #$20
8040-   8D 5A 80    STA   $805A
8043-   4C 4B 80    JMP   $804B
8046-   A9 40       LDA   #$40
8048-   8D 5A 80    STA   $805A
804B-   A5 08       LDA   $08
804D-   85 06       STA   $06
804F-   A5 09       LDA   $09
8051-   85 07       STA   $07
8053-   A6 1F       LDX   $1F
8055-   BD 00 81    LDA   $8100,X
8058-   18          CLC
8059-   69 40       ADC   #$40
805B-   85 09       STA   $09
805D-   BD 00 82    LDA   $8200,X
8060-   85 08       STA   $08
8062-   A0 27       LDY   #$27
8064-   B1 08       LDA   ($08),Y
8066-   91 06       STA   ($06),Y
8068-   88          DEY
8069-   10 F9       BPL   $8064
806B-   A0 00       LDY   #$00
806D-   68          PLA
```

```
806E-   91 08      STA   ($08),Y
8070-   C8         INY
8071-   C0 28      CPY   #$28
8073-   90 F8      BCC   $806D
8075-   E6 1F      INC   $1F
8077-   A5 1F      LDA   $1F
8079-   C9 C0      CMP   #$C0
807B-   D0 8B      BNE   $8008
807D-   E6 1E      INC   $1E
807F-   A5 1E      LDA   $1E
8081-   C9 02      CMP   #$02
8083-   D0 01      BNE   $8086
8085-   60         RTS
8086-   4C 04 80   JMP   $8004
8089-   A9 20      LDA   #$20
808B-   85 E6      STA   $E6
808D-   A2 00      LDX   #$00
808F-   8A         TXA
8090-   48         PHA
8091-   20 11 F4   JSR   $F411
8094-   68         PLA
8095-   AA         TAX
8096-   A5 26      LDA   $26
8098-   9D 00 82   STA   $8200,X
809B-   A5 27      LDA   $27
809D-   9D 00 81   STA   $8100,X
80A0-   E8         INX
80A1-   E0 C0      CPX   #$C0
80A3-   D0 EA      BNE   $808F
80A5-   AD 54 C0   LDA   $C054
80A8-   AD 50 C0   LDA   $C050
80AB-   AD 57 C0   LDA   $C057
80AE-   AD 52 C0   LDA   $C052
80B1-   AD 10 C0   LDA   $C010
80B4-   20 00 80   JSR   $8000
80B7-   AD 00 C0   LDA   $C000
80BA-   10 F5      BPL   $80B1
80BC-   60         RTS
```

$8038—$804A　　移完后，取$1E单元内容，查看是图象第几页，以决定后一页图象的地址，当$1E为0时，高位地址加$20取第二页地址，当$1E为1时，高位地址加$40取第三页地址。

$804B—$807C　　同子程序GP2B01，完成第二页和第三页图象上移显示。

$807D—$8084　　页数计数器加1用以显示第三页。

$8085　　全部移完返回。

$8086—$80BC　　演示用。

演示子程序GP2B04，可用程序GP2B05。

```
10 D$ =  CHR$ (4)
20  PRINT D$"BLOAD PIC-1,A$2000"
30  PRINT D$"BLOAD PIC-2,A$4000"
35  PRINT D$"BLOAD PIC-3,A$6000"
40  PRINT D$"BLOAD GP2B04"
50  CALL 32905
60  FOR I = 1 TO 1000: NEXT I
70  TEXT : HOME : END
```

要将三页图象向下移动，同GP2B01改动为GP2B03。

```
8000-    A9 00       LDA   #$00
8002-    85 1E       STA   $1E
8004-    A9 BF       LDA   #$BF
8006-    85 1F       STA   $1F
8008-    A2 BF       LDX   #$BF
800A-    BD 00 81    LDA   $8100,X
800D-    85 07       STA   $07
800F-    BD 00 82    LDA   $8200,X
8012-    85 06       STA   $06
8014-    CA          DEX
```

```
8015-    BD 00 81    LDA   $8100,X
8018-    85 09       STA   $09
801A-    BD 00 82    LDA   $8200,X
801D-    85 08       STA   $08
801F-    E0 BE       CPX   #$BE
8021-    D0 08       BNE   $802B
8023-    A0 27       LDY   #$27
8025-    B1 06       LDA   ($06),Y
8027-    48          PHA
8028-    88          DEY
8029-    10 FA       BPL   $8025
802B-    A0 27       LDY   #$27
802D-    B1 08       LDA   ($08),Y
802F-    91 06       STA   ($06),Y
8031-    88          DEY
8032-    10 F9       BPL   $802D
8034-    E0 00       CPX   #$00
8036-    D0 D2       BNE   $800A
8038-    A5 1E       LDA   $1E
803A-    C9 00       CMP   #$00
803C-    D0 08       BNE   $8046
803E-    A9 20       LDA   #$20
8040-    8D 5A 80    STA   $805A
8043-    4C 4B 80    JMP   $804B
8046-    A9 40       LDA   #$40
8048-    8D 5A 80    STA   $805A
804B-    A5 08       LDA   $08
804D-    85 06       STA   $06
804F-    A5 09       LDA   $09
8051-    85 07       STA   $07
8053-    A6 1F       LDX   $1F
8055-    BD 00 81    LDA   $8100,X
8058-    18          CLC
8059-    69 40       ADC   #$40
805B-    85 09       STA   $09
805D-    BD 00 82    LDA   $8200,X
8060-    85 08       STA   $08
```

```
8062-   A0 27      LDY   #$27
8064-   B1 08      LDA   ($08),Y
8066-   91 06      STA   ($06),Y
8068-   88         DEY
8069-   10 F9      BPL   $8064
806B-   A0 00      LDY   #$00
806D-   68         PLA
806E-   91 08      STA   ($08),Y
8070-   C8         INY
8071-   C0 28      CPY   #$28
8073-   90 F8      BCC   $806D
8075-   C6 1F      DEC   $1F
8077-   A5 1F      LDA   $1F
8079-   C9 FF      CMP   #$FF
807B-   D0 8B      BNE   $8008
807D-   E6 1E      INC   $1E
807F-   A5 1E      LDA   $1E
8081-   C9 02      CMP   #$02
8083-   D0 01      BNE   $8086
8085-   60         RTS
8086-   4C 04 80   JMP   $8004
8089-   A9 20      LDA   #$20
808B-   85 E6      STA   $E6
808D-   A2 00      LDX   #$00
808F-   8A         TXA
8090-   48         PHA
8091-   20 11 F4   JSR   $F411
8094-   68         PLA
8095-   AA         TAX
8096-   A5 26      LDA   $26
8098-   9D 00 82   STA   $8200,X
809B-   A5 27      LDA   $27
809D-   9D 00 81   STA   $8100,X
80A0-   E8         INX
80A1-   E0 C0      CPX   #$C0
80A3-   D0 EA      BNE   $808F
80A5-   AD 54 C0   LDA   $C054
```

```
80A8-    AD 50 C0      LDA   $C050
80AB-    AD 57 C0      LDA   $C057
80AE-    AD 52 C0      LDA   $C052
80B1-    AD 10 C0      LDA   $C010
80B4-    20 00 80      JSR   $8000
80B7-    AD 00 C0      LDA   $C000
80BA-    10 F5         BPL   $80B1
80BC-    60            RTS
```

一样，只要将图象从末行开始，即将末行地址的单元内容先压栈，随后递减倒着取地址即可。见程序GP2B06。

我们可以用图象上、下移动显示的原理，对子程序进行改进，使之变化出更多的形式来。程序GP2B07就是改进后的一例。运行这一程序，我们可以将第一页图象中的一部分向上移动显示，也可全部向上移动显示，还可按照需要来确定移动显示一页、二页或三页的图形，这样就灵活多了。

这个程序设置了3个单元给用户，$01单元用来存放第一页图象被移启始行，$02单元用来存放第一页图象被移结束行，$1E单元用来存放需移动的图象页数并加1。

程序原理说明如下：

$8000—$8007　初始化，设置计数器$1F初值为0，并确定图象显示第一页

$8008—$800C　计算被移启始行地址。

$800D—$8014　将启始行压栈。

程序GP2B07清单。

```
8000-    A9 00         LDA   #$00
8002-    85 1F         STA   $1F
8004-    A9 20         LDA   #$20
8006-    85 E6         STA   $E6
8008-    A5 01         LDA   $01
```

```
800A-   20 11 F4    JSR   $F411
800D-   A0 27       LDY   #$27
800F-   B1 26       LDA   ($26),Y
8011-   48          PHA
8012-   88          DEY
8013-   10 FA       BPL   $800F
8015-   E6 1F       INC   $1F
8017-   A5 1F       LDA   $1F
8019-   C5 1E       CMP   $1E
801B-   F0 50       BEQ   $806D
801D-   C9 01       CMP   #$01
801F-   D0 0E       BNE   $802F
8021-   A5 01       LDA   $01
8023-   85 03       STA   $03
8025-   A5 02       LDA   $02
8027-   85 04       STA   $04
8029-   20 47 80    JSR   $8047
802C-   4C 15 80    JMP   $8015
802F-   A5 E6       LDA   $E6
8031-   18          CLC
8032-   69 20       ADC   #$20
8034-   85 E6       STA   $E6
8036-   20 90 80    JSR   $8090
8039-   A9 00       LDA   #$00
803B-   20 11 F4    JSR   $F411
803E-   20 63 80    JSR   $8063
8041-   20 47 80    JSR   $8047
8044-   4C 15 80    JMP   $8015
8047-   20 90 80    JSR   $8090
804A-   A5 03       LDA   $03
804C-   20 11 F4    JSR   $F411
804F-   20 63 80    JSR   $8063
8052-   E6 03       INC   $03
8054-   A5 03       LDA   $03
8056-   C5 04       CMP   $04
8058-   D0 ED       BNE   $8047
805A-   A9 00       LDA   #$00
```

```
805C-    85 03       STA   $03
805E-    A9 C0       LDA   #$C0
8060-    85 04       STA   $04
8062-    60          RTS
8063-    A0 27       LDY   #$27
8065-    B1 26       LDA   ($26),Y
8067-    91 08       STA   ($08),Y
8069-    88          DEY
806A-    10 F9       BPL   $8065
806C-    60          RTS
806D-    A0 00       LDY   #$00
806F-    68          PLA
8070-    91 26       STA   ($26),Y
8072-    C8          INY
8073-    C0 28       CPY   #$28
8075-    D0 F8       BNE   $806F
8077-    60          RTS
8078-    AD 54 C0    LDA   $C054
807B-    AD 50 C0    LDA   $C050
807E-    AD 57 C0    LDA   $C057
8081-    AD 52 C0    LDA   $C052
8084-    AD 10 C0    LDA   $C010
8087-    20 00 80    JSR   $8000
808A-    AD 00 C0    LDA   $C000
808D-    10 F5       BPL   $8084
808F-    60          RTS
8090-    A5 26       LDA   $26
8092-    85 08       STA   $08
8094-    A5 27       LDA   $27
8096-    85 09       STA   $09
8098-    60          RTS
```

$8015—$801C　判断全部图象移完否，移完转$806D，未移完往下执行。

$801D—$8020　判断是否刚处理过首行图形，是，

往下继续执行，否，则跳到$802F开始执行。

$8021—$8028　将$01、$02单元内容送往$03、$04单元，用$03、$04单元参与变化，从而保存$01、$02单元内容。

$8029—$802E　调子程序$8047—$8062，将一页内的图象全部移完，并转$8015继续。

$802F—$8038　将图象显示定为第二页，或第三页，并将目的行地址放入$08、$09单元中。

$8039—$803D　计算下一页(第二页或第三页)首行地址。

$803E—$8040　调子程序$8063—$806C，移动一行图象。

$8041—$8046　同$8029—$802E。

$8047—$8062　将一页内的图象全部移完子程序。

其中：

$8047—$804B　取目的行地址放入$08、$09单元。

$804A—$804E　计算被移行地址。

$804F—$8051　调子程序$8063—$806C，移动一行图象。

$8052—$8059　$03单元内容加1，判断是否移完图象，未完转回$8047继续移。

$805A—$8061　移完，将$03单元置0，$04单元置$C0，为下一页图象移动作好准备。

$8062　　　　　　结束返回。

$8063—$806C　　移动一行子程序。

$806D—$8077　　全部图象移完后，将压栈的首行弹出，整个程序结束。

$8078—$808F　　演示用。

$8090—$8098　　取目的行地址送入$08，$09单元子程序。

用程序GP2B08演示程序GP2B07。

```
10 D$ = CHR$ (4)
20  PRINT D$"BLOAD PIC-1,A$2000"
30  PRINT D$"BLOAD GP2B07"
40  POKE 1,80: POKE 2,112: POKE 30
    ,2
50  CALL 32888
60  FOR I = 1 TO 1000: NEXT I
70  TEXT : HOME : END
```

执行程序GP2B08，演示的是第一页图象的中间一部分($50—$70行)向上移动显示，如果要演示全部，则只要将40句改为：

40 POKE 1，0：POKE 2，192：POKE30，4

即可，当然，三页图象也必须先调入内存。

第三节　多页图象左移联动显示

类似子程序GP2B07，这里给出一个可多页左移显示，或全幅或部分图象移动的子程序GP2B09。其移动显示原理见图2.73，程序编写不同GP2610显示效果也不如GP2610这个程序也给用户三个单元，$01单元用来存放被移图象起始行，$02单元用来存放被移图象终止行，$1E单元用来存

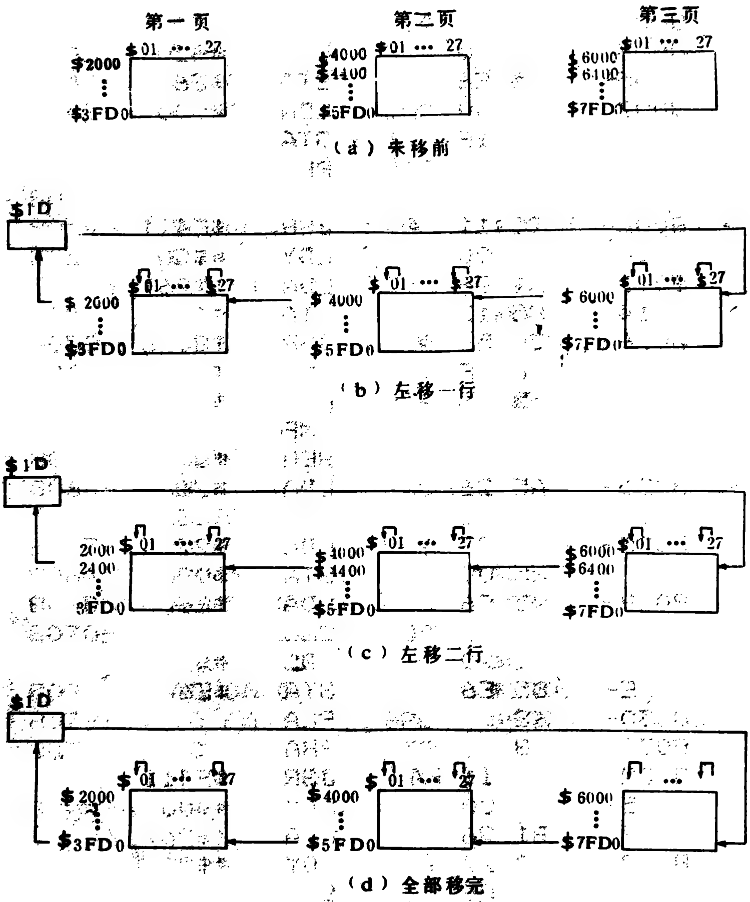

第一页
第二页
第三页
$01 … 27
$2000
$3FD0
$4000
$4400
$5FD0
$6000
$6400
$7FD0
（a）未移前
$1D
$01 … 27
$2000
$3FD0
$4000
$5FD0
$6000
$7FD0
（b）左移一行
$1D
2000
2400
3FD0
$4000
$4400
$5FD0
$6000
$6400
$7FD0
（c）左移二行
$1D
$2000
$3FD0
$4000
$5FD0
$6000
$7FD0
（d）全部移完

图 2.73

GP.2 B 09清单。

```
8000-   A5 01       LDA   $01
8002-   48          PHA
8003-   A9 20       LDA   #$20
8005-   85 E6       STA   $E6
8007-   A9 00       LDA   #$00
8009-   85 1F       STA   $1F
800B-   68          PLA
800C-   48          PHA
800D-   20 11 F4    JSR   $F411
8010-   A0 00       LDY   #$00
8012-   B1 26       LDA   ($26),Y
8014-   85 1D       STA   $1D
8016-   20 56 80    JSR   $8056
8019-   E6 1F       INC   $1F
801B-   A5 1F       LDA   $1F
801D-   C5 1E       CMP   $1E
801F-   F0 23       BEQ   $8044
8021-   A5 26       LDA   $26
8023-   85 08       STA   $08
8025-   A5 27       LDA   $27
8027-   85 09       STA   $09
8029-   A5 E6       LDA   $E6
802B-   18          CLC
802C-   69 20       ADC   #$20
802E-   85 E6       STA   $E6
8030-   68          PLA
8031-   48          PHA
8032-   20 11 F4    JSR   $F411
8035-   A0 00       LDY   #$00
8037-   B1 26       LDA   ($26),Y
8039-   A0 27       LDY   #$27
803B-   91 08       STA   ($08),Y
803D-   A0 00       LDY   #$00
803F-   20 56 80    JSR   $8056
8042-   F0 D5       BEQ   $8019
8044-   A0 27       LDY   #$27
```

```
8046-    A5 1D       LDA   $1D
8048-    91 26       STA   ($26),Y
804A-    68          PLA
804B-    18          CLC
804C-    69 01       ADC   #$01
804E-    C5 02       CMP   $02
8050-    D0 01       BNE   $8053
8052-    60          RTS
8053-    48          PHA
8054-    D0 AD       BNE   $8003
8056-    C8          INY
8057-    C0 28       CPY   #$28
8059-    F0 08       BEQ   $8063
805B-    B1 26       LDA   ($26),Y
805D-    88          DEY
805E-    91 26       STA   ($26),Y
8060-    C8          INY
8061-    D0 F3       BNE   $8056
8063-    60          RTS
8064-    AD 54 C0    LDA   $C054
8067-    AD 50 C0    LDA   $C050
806A-    AD 57 C0    LDA   $C057
806D-    AD 52 C0    LDA   $C052
8070-    AD 10 C0    LDA   $C010
8073-    20 00 80    JSR   $8000
8076-    AD 00 C0    LDA   $C000
8079-    10 F5       BPL   $8070
807B-    60          RTS
```

放需移动的图象页数。

程序原理说明如下：

$8000—$8002　将$01单元内容压栈保存。

$8003—$8006　置图象显示第一页。

$8007—$800A　计数单元$1F置0。

$800B—$800F　计算被移起始行地址。

$8010—$8015　将起始行0列送$1D单元暂存，见图2.73（b）。

$8016—$8018　调左移一行子程序左移一行，见图2.73(b)。

$8019—$8020　判断需要移动的图象在设定的页数内是否移完，移完转$8044，未移完往下执行。

$8021—$8028　将前页末列($27列)地址送$08、$09单元，作为目的行地址。

$8029—$802F　置图象显示第二、第三页。

$8030—$8034　计算下一页被移行地址。

$8035—$803C　将下页0列左移送往前页$27列，见图2.73（b)。

$803D—$8041　调左移一行子程序左移一行。

$8042—$8043　转$8019再作判断。

$8044—$8049　当图象在指定页数内移完后，将暂存在$1D单元的内容送往末列，形成图象绕卷，见图2.73（b)。

$804A—$8051　当一行左移完后，从栈内弹出$01单元内容并加1，再与设定的$02单元内容比较，判全部图象是否移完，如未移完，则转$8053。

$8052　全部移完结束。

$8053—$8055　将计数内容压栈，转$8003再移，见图2.73（c)，2.73(d)。

$8056—$8063　左移一行子程序。

其中：

$8056—$805A　先判一行是否移完，移完转$8063返回，未移完往下执行继续移。

$805B—$805F　左移一列。

$8060—$8062　列+1，转$8056判断。

$8063　返回。

$8064—$807B　演示用。

演示程序GP2B09，可将GP2B08程序中30句和50句改为：

```
30    PRINT  D$;"BLOAD  GP2B09"
50    CALL  32868
```

其余不变即可。

根据上面的分析，相信读者在此基础上完全可以将程序改为向右移动显示子程序，这里就不再多说了。

第十二章　特殊技巧

中华学习机（包括紫金Ⅱ及APPLE Ⅱ）内存比较小，这在运行较大程序特别是处理多幅图形时，常常感到不便，为了充分利用有限的内存空间，常常需要采取一些特殊技巧，而巧妙地利用16K RAM语言卡和图象区缩存贮技术，可以解决这方面的问题。本章就向读者详细介绍16K RAM语言卡和压缩存贮技术，并提供了使用和操作实例。读者可以利用本章提供的技巧和方法，引用到您的软件中，相信会在图形处理辅助设计、计算机辅助教学、游戏软件开发、办公室自动化诸方面，得到更为广泛的应用。

第一节　16K RAM卡简介

前面几章谈到的图象显示技巧，多数是在高分辨率第一、第二页进行的，最多也只能显示三页图象，这是由于中华学习机的内存限制所造成的。而实际上，中华学习机还有16K RAM语言卡，由于地址和ROM重复，一般情况下都空着未用，因此我们可以将其用起来，这样图象可增至5页。这里，

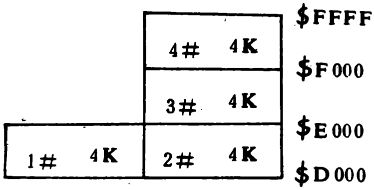


图 2.74　16K RAM地址分配图

我们介绍一个程序，可使5页图象上移绕卷。在介绍程序前，我们先简单介绍一下如何使用中华学习机空着的16K RAM。

中华学习机16K RAM地址分配如图2.74，由于它的地址与ROM是重复的，因此要用软开关切换的办法来选通它，在16K RAM中，1#4K体和2#4K体地址也是重复的，也

程序GP2C01清单。

```
8300-    A9 06       LDA   #$06
8302-    8D 39 83    STA   $8339
8305-    8D 58 83    STA   $8358
8308-    A9 08       LDA   #$08
830A-    8D 3B 83    STA   $833B
830D-    8D 5A 83    STA   $835A
8310-    4C 23 83    JMP   $8323
8313-    A9 06       LDA   #$06
8315-    8D 3B 83    STA   $833B
8318-    8D 5A 83    STA   $835A
831B-    A9 08       LDA   #$08
831D-    8D 39 83    STA   $8339
8320-    8D 58 83    STA   $8358
8323-    AE 83 C0    LDX   $C083
8326-    AE 83 C0    LDX   $C083
8329-    A9 D0       LDA   #$D0
832B-    85 07       STA   $07
832D-    A9 20       LDA   #$20
832F-    85 09       STA   $09
8331-    A9 00       LDA   #$00
8333-    85 06       STA   $06
8335-    85 08       STA   $08
8337-    A8          TAY
8338-    B1 06       LDA   ($06),Y
833A-    91 08       STA   ($08),Y
833C-    88          DEY
833D-    D0 F9       BNE   $8338
833F-    E6 07       INC   $07
8341-    E6 09       INC   $09
```

```
8343-    A5 09        LDA   $09
8345-    C9 50        CMP   #$50
8347-    D0 EF        BNE   $8338
8349-    AE 8B C0     LDX   $C08B
834C-    AE 8B C0     LDX   $C08B
834F-    A9 D0        LDA   #$D0
8351-    85 07        STA   $07
8353-    A9 50        LDA   #$50
8355-    85 09        STA   $09
8357-    B1 06        LDA   ($06),Y
8359-    91 08        STA   ($08),Y
835B-    88           DEY
835C-    D0 F9        BNE   $8357
835E-    E6 07        INC   $07
8360-    E6 09        INC   $09
8362-    A5 09        LDA   $09
8364-    C9 60        CMP   #$60
8366-    D0 EF        BNE   $8357
8368-    AD 81 C0     LDA   $C081
836B-    60           RTS
```

要用软开关来分别选用。我们可用指令 LDX ＄C 083 来接通1#、3#、4#3个4K体，并置这12K空间为可读/写的状态。用LDX ＄C08B 指令来接通2#、3#、4# 3个4K体，并置这12K空间为可读/写状态。在选通上述RAM时，ROM将不起作用，因而在程序结束时还要用指令LDX ＄C081来切断RAM，恢复ROM。至于其它软开关的设置，请看其它书籍介绍，这里就不详谈了。

第二节 16K RAM卡使用

用上述 3 个指令，我们就可以将图象资料写入 RAM 体并可读出了。程序GP 2C01就可以将两页图形资料写入

RAM体，也可从RAM体中读出。

下面分析一下原理：

$8300—$8312　　初始化，置程序为读出状态，转$8323开始执行。

$8313—$8322　　初始化，置程序为写入状态。

$8323—$8328　　读两次$C083，选通1#、3#、4# 3个4K体，并置为可读/写状态。

$8329—$8336　　将图象第一页首地址$2000送入$09、$08单元，将1#4K体首地址$D000送入$07，$06单元。

$8337—$833E　　开始逐列传送（读或写）。

$833F—$8348　　高八位地址加1，判断12K是否全部读出（或写入），未完转$8038继续。

$8349—$834E　　12K读完（或写完），再选通2#、3#、4# 3个4K体，并置为读/写状态。

$834F—$8356　　将2#4K体首地址高八位送$07单元，将第2页后4K地址高八位送$09单元。

$8357—$835D　　继续逐列传送（读或写）。

$835E—$8367　　高八位地址加1，判断4K是否全部读出（或写入），未完转$8357继续。

$8368—$836A　　4K读完（或写完），则切断RAM体，恢复ROM。

$836B　　结束，返回。

使用本程序，要先将两页图形存放于第一、第二页中，然后调用该程序先写入，后读出。要演示程序GP 2 C01，可

用GP 2C02程序。

```
10 D$ =  CHR$ (4)
20  PRINT D$"BLOAD PIC-1,A$2000"
30  PRINT D$"BLOAD PIC-2,A$4000"
40  PRINT D$"BLOAD GP2C01"
50  CALL 33555
60  HGR : HGR2 : TEXT
70  CALL 33536
80  POKE  - 16297,0: POKE  - 16300
    ,0: POKE  - 16302,0: POKE  -
    16304,0
90  FOR I = 1 TO 2000: NEXT I
100  POKE  - 16299,0
110  FOR I = 1 TO 2000: NEXT I
120  TEXT : HOME : END
```

第三节　五页图象绕卷显示

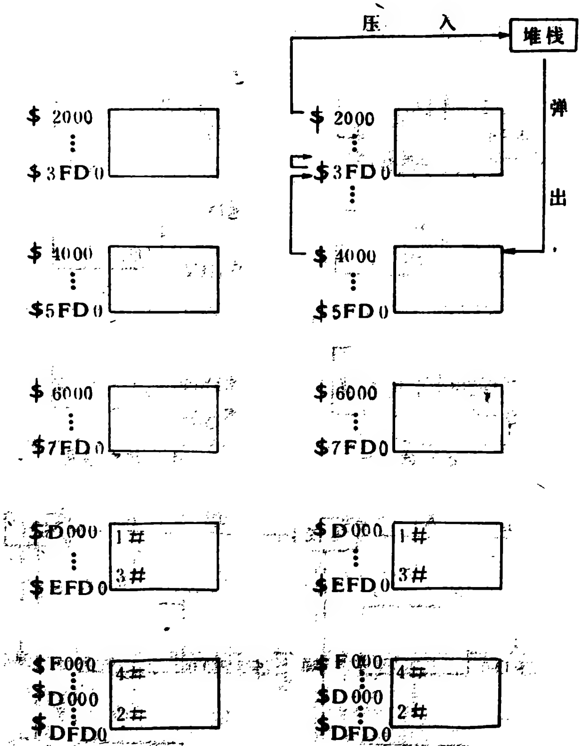
搞清了RAM体的使用，我们再介绍五页图象上移绕卷程序。程序见GP2C03，其原理见图2.75。

程序GP2C03原理说明如下：

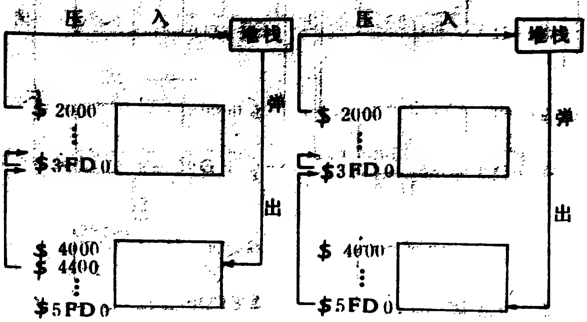
$8000—$8003　　图象页数计数器$1E置0。$1E内容供程序判别用，为0时，取第二页地址；为1时，取第三页地址；为2时取第四页地址；为3时，取第五页地址。

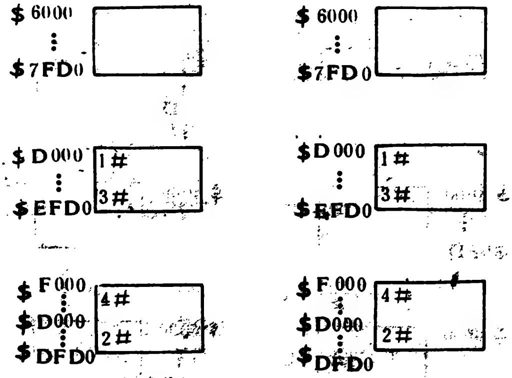
$8004—$8013　　查表，取移动后目的行地址，高八位放入$07单元，低八位放入$06单元。

$8014—$801E　　查表指针加1后，取得被移行地址，高八位放入$09单元，低八位放入$08单元。


$2000
$3FD0
$4000
$5FD0
$6000
$7FD0
$D000
1#
3#
$EFD0
$F000
4#
$D000
2#
$DFD0
压
入
堆栈
弹
出
$2000
$3FD0
$4000
$5FD0
$6000
$7FD0
$D000
1#
3#
$EFD0
$F000
4#
$D000
2#
$DFD0

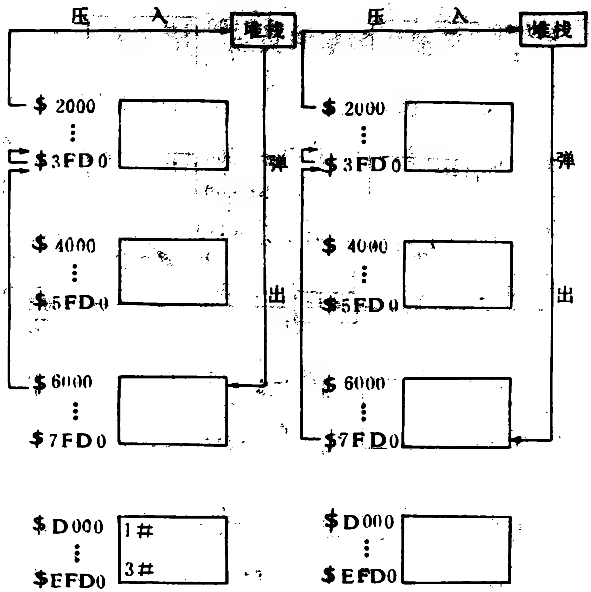
（a）未移　　　　（b）移一行


压
入
堆栈
弹
出
$2000
$3FD0
$4000
$4400
$5FD0
压
入
堆栈
弹
出
$2000
$3FD0
$4000
$5FD0

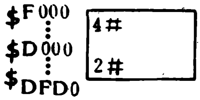


$6000
$7FD0
$D000
$EFD0
1#
3#
$F000
$D000
$DFD0
4#
2#
$6000
$7FD0
$D000
$EFD0
1#
3#
$F000
$D000
$DFD0
4#
2#

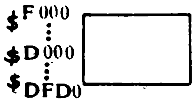
（c）移二行　　（d）两页移完



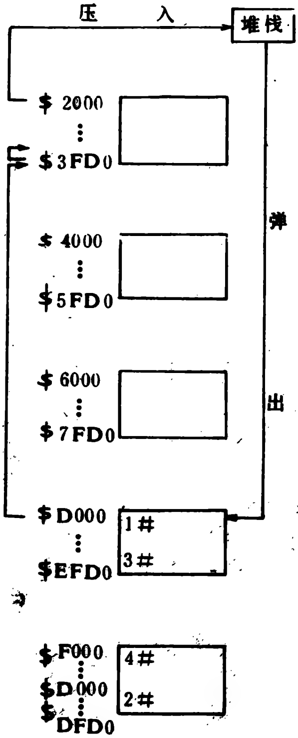
压入
堆栈
弹出
$2000
$3FD0
$4000
$5FD0
$6000
$7FD0
$D000
$EFD0
1#
3#
压入
堆栈
弹出
$2000
$3FD0
$4000
$5FD0
$6000
$7FD0
$D000
$EFD0



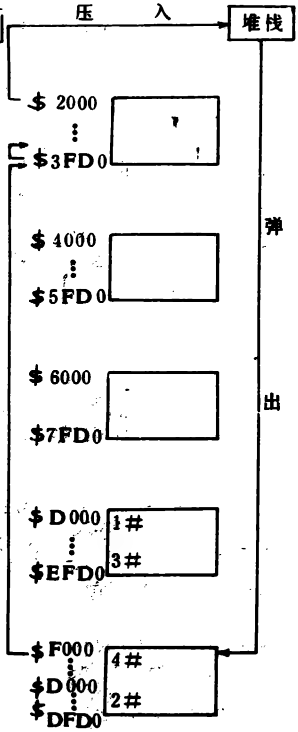
（e）移第三页



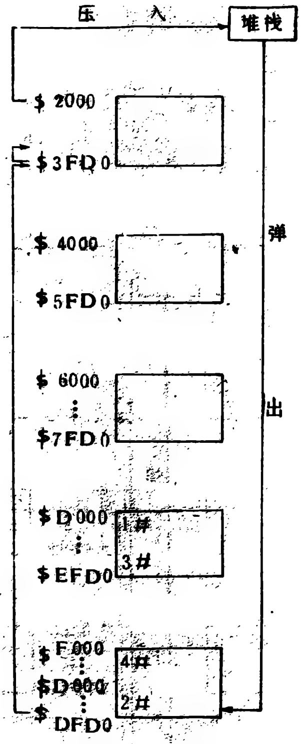
（f）三页移完



（g）移第四页



（h）移第五页



（i）第五页移完

图 2.75

程序GP2C03清单。

```
8000-    A9 00       LDA   #$00
8002-    85 1E       STA   $1E
8004-    A9 00       LDA   #$0D
8006-    85 1F       STA   $1F
8008-    A2 00       LDX   #$00
800A-    BD 00 81    LDA   $8100,X
800D-    85 07       STA   $07
```

```
800F-    BD 00 82    LDA   $8200,X
8012-    85 06       STA   $06
8014-    E8          INX
8015-    BD 00 81    LDA   $8100,X
8018-    85 09       STA   $09
801A-    BD 00 82    LDA   $8200,X
801D-    85 08       STA   $08
801F-    E0 01       CPX   #$01
8021-    D0 08       BNE   $802B
8023-    A0 27       LDY   #$27
8025-    B1 06       LDA   ($06),Y
8027-    48          PHA
8028-    88          DEY
8029-    10 FA       BPL   $8025
802B-    A0 27       LDY   #$27
802D-    B1 08       LDA   ($08),Y
802F-    91 06       STA   ($06),Y
8031-    88          DEY
8032-    10 F9       BPL   $802D
8034-    E0 BF       CPX   #$BF
8036-    D0 D2       BNE   $800A
8038-    A5 1E       LDA   $1E
803A-    C9 00       CMP   #$00
803C-    D0 08       BNE   $8046
803E-    A9 20       LDA   #$20
8040-    8D 8F 80    STA   $808F
8043-    4C 6F 80    JMP   $806F
8046-    C9 01       CMP   #$01
8048-    D0 08       BNE   $8052
804A-    A9 40       LDA   #$40
804C-    8D 8F 80    STA   $808F
804F-    4C 6F 80    JMP   $806F
8052-    C9 02       CMP   #$02
8054-    D0 0E       BNE   $8064
8056-    AD 83 C0    LDA   $C083
8059-    AD 83 C0    LDA   $C083
805C-    A9 B0       LDA   #$B0
```

```
805E-   8D 8F 80    STA   $808F
8061-   4C 6F 80    JMP   $806F
8064-   AD 8B C0    LDA   $C08B
8067-   AD 8B C0    LDA   $C08B
806A-   A9 D0       LDA   #$D0
806C-   8D 8F 80    STA   $808F
806F-   A5 08       LDA   $08
8071-   85 06       STA   $06
8073-   A5 09       LDA   $09
8075-   85 07       STA   $07
8077-   A6 1F       LDX   $1F
8079-   BD 00 81    LDA   $8100,X
807C-   A4 1E       LDY   $1E
807E-   C0 03       CPY   #$03
8080-   D0 0B       BNE   $808D
8082-   C9 30       CMP   #$30
8084-   90 07       BCC   $808D
8086-   48          PHA
8087-   A9 A0       LDA   #$A0
8089-   8D 8F 80    STA   $808F
808C-   68          PLA
808D-   18          CLC
808E-   69 A0       ADC   #$A0
8090-   85 09       STA   $09
8092-   BD 00 82    LDA   $8200,X
8095-   85 08       STA   $08
8097-   A0 27       LDY   #$27
8099-   B1 08       LDA   ($08),Y
809B-   91 06       STA   ($06),Y
809D-   88          DEY
809E-   10 F9       BPL   $8099
80A0-   A0 00       LDY   #$00
80A2-   68          PLA
80A3-   91 08       STA   ($08),Y
80A5-   C8          INY
80A6-   C0 28       CPY   #$28
```

```
80A8-   90 F8       BCC   $80A2
80AA-   E6 1F       INC   $1F
80AC-   A5 1F       LDA   $1F
80AE-   C9 C0       CMP   #$C0
80B0-   D0 0F       BNE   $80C1
80B2-   E6 1E       INC   $1E
80B4-   A5 1E       LDA   $1E
80B6-   C9 04       CMP   #$04
80B8-   D0 04       BNE   $80BE
80BA-   AD 81 C0    LDA   $C081
80BD-   60          RTS
80BE-   4C 04 80    JMP   $8004
80C1-   4C 08 80    JMP   $8008
80C4-   A9 20       LDA   #$20
80C6-   85 E6       STA   $E6
80C8-   A2 00       LDX   #$00
80CA-   8A          TXA
80CB-   48          PHA
80CC-   20 11 F4    JSR   $F411
80CF-   68          PLA
80D0-   AA          TAX
80D1-   A5 26       LDA   $26
80D3-   9D 00 82    STA   $8200,X
80D6-   A5 27       LDA   $27
80D8-   9D 00 81    STA   $8100,X
80DB-   E8          INX
80DC-   E0 C0       CPX   #$C0
80DE-   D0 EA       BNE   $80CA
80E0-   AD 54 C0    LDA   $C054
80E3-   AD 50 C0    LDA   $C050
80E6-   AD 57 C0    LDA   $C057
80E9-   AD 52 C0    LDA   $C052
80EC-   AD 10 C0    LDA   $C010
80EF-   20 00 80    JSR   $8000
80F2-   AD 00 C0    LDA   $C000
80F5-   10 F5       BPL   $80EC
80F7-   60          RTS
```

$801F—$802A　判断是否为首行，如是则将首行逐列压入堆栈，如图2.75(b)，如不是则跳过此段转$802B往下执行。

$802B—$8033　将一行逐列向上移动。

$8034—$8037　判断整幅画面是否全部上移了一行，未移完则转至$800A继续移。

$8038—$803D　移完后，取$1E单元内容，查看是否为0，不为0转$8040再查。

$803E—$8045　为0，则将#$20送入$808F，以便取第二页地址，同时转$806F执行，见图2.75(b)。

$8046—$8049　再查$1E是否为#$01，不是转$8052再查。

$804A—$8051　为1，则将#$40送入$808F，以便取第三页地址，同时转$806F执行，见图2.75(e)。

$8052—$8055　再查$1E是否为#$02，不是转$8064再查。

$8056—$8063　为2，则选通RAM1#、3#、4#3个4K体，置为读/写状态，并将#$B0送入$808F，以便取第四页地址，同时转$806F执行，见图2.75(g)。

$8064—$806E　为3，则选通RAM2#、3#、4#3个4K体，置为读/写状态，并将#$D0送入$808F，以便取第五页地址，见图2.75(h)。

$8069F—$8076　　将上页末行地址作为目的行地址。

$8077—$807B　　取高八位地址。

$807C—$8081　　判$1E是否为#$03，不为3转$808D。

$8082—$8085　　判高八位地址是否小于$30，是，转$808D。

$8086—$808C　　不是，则将高八位地址压栈保存，将#$A0送$808F，以便取第五页2#4K体地址，然后弹出高八位地址。

$808D—$8096　　根据$1E内容分别取后一页地址送$09，$08单元。

$8097—$809F　　传送一行，见图2.75(b)，(e)，(g)，(h)。

$80A0—$80A9　　堆栈弹出一行，见图2.75(b)，(e)，(g)，(h)。

$80A0—$80B1　　判断第二、第三、第四、第五页是否移完，未完转$8008继续，见图2.75(c)、(d)、(f)、(i)。

$80B2—$80B9　　移完，再判是否全部移完，未完转$8004继续移。

$80BA—$80BC　　全部移完，切断RAM体，恢复ROM。

$80BD　　结束，返回。

$80C4—$80DF　　计算各行首地址。

$80E0—$80F7　　演示用。

用程序GP2C04演示程序GP2C03。

```
10 D$ =  CHR$ (4)
20  PRINT D$"BLOAD PIC-1,A$2000"
30  PRINT D$"BLOAD PIC-2,A$4000"
40  PRINT D$"BLOAD GP2C01"
50  CALL 33555
60  PRINT D$"BLOAD PIC-3,A$2000"
70  PRINT D$"BLOAD PIC-4,A$4000"
80  PRINT D$"BLOAD PIC-5,A$6000"
90  PRINT D$"BLOAD GP2C03"
100  CALL 32964
110  FOR I = 1 TO 2000: NEXT I
120  TEXT : HOME : END
```

第四节　压缩存贮简介

存贮在$1000—$10FA中的机器语言程序GP2C05，能够完成对图形的压缩和还原功能。例如一幅具有34个扇区占用8K存贮空间的图形，经GP2C05程序处理后，再存盘时只占用不到10个扇区3K左右的存贮空间。同时，这种经过压缩存贮的图形资料，仍用GP2C05程序处理，还原成真正的原样图形。这就是压缩存贮技术。利用压缩存贮技术，将使中华学习机有限的存贮空间，得到更加充分的利用，对处理多幅画面的分页显示及多页图形的联动显示，提供有力的工具。

为说明这种压缩存贮和还原显示图形的功能，下面详细介绍操作步骤。

1. 图形压缩

·BLOAD GP2C05 ↙

·BLOAD C1,A$2000↙

·CALL −151↙

·*0:00　60↙

·*E6:20↙

·*1000G↙

·*0.1↙

·0000—5E　67

·BSAVE　TC1，A$6000，L$75F↙

其中，GP2C05是压缩还原程序，它是以 BSAVE GP2C05 A$1000，L$FB存于磁盘中，供实用时调用的。

C1是被处理的图形，原来占用34扇区8K存贮空间，在处理时（这里是指压缩）先调入$2000到$3FFF的第一页。

进入监控后，在$00—$01单元键入00　60是指将要压缩的图形压缩后存放在$6000开始的单元中。

在$E6单元中键入20，是指被处理的图形，在处理前存放在$2000中。

*1000G↙，就是对C1图形进行压缩处理，而处理后的压缩图形存放在$6000开始的单元中。

检查监控状态下$00和$01单元，显示出0000—5E67，则表示压缩后的图形其长度为$675E—$6000＝$75E。

所以，重新存贮压缩处理后的图形，应重新命名，例如本例取TC1，并将它以BSAVE　TC1，A$6000，L$75F存于盘中，表明压缩后的资料存放在$6000开始的单元中，其长度是$75E（用L$75F是防止丢“尾巴”）。

2．还原显示

·BLOAD　TC1↙

·CALL　—151↙

· *0:00　60↙

· *E6:20↙

· *1003G↙

· RUN　GP2C06↙

其中，BLOAD　TC1↙是调压缩存贮的图形资料进内存，$00单元、$01单元、$E6单元键入的值同图形压缩中的值，而1003G↙则是执行还原图形的功能，最后，在BASIC状态下，运行第一页图形的显示程序GP2C06，即可看到还原好的图象。

压缩还原程序GP2C05清单。

```
1000-  4C  7F  10  A5  E6  09  04  85
1008-  06  A2  01  86  04  A0  00  84
1010-  02  84  05  84  08  B1  00  D0
1018-  18  E6  00  D0  02  E6  01  B1
1020-  00  85  08  E6  00  D0  02  E6
1028-  01  B1  00  85  07  A5  07  C6
1030-  08  A4  02  91  05  E8  E8  E0
1038-  BF  90  12  E6  02  A4  02  C0
1040-  28  90  08  C6  04  30  15  A0
1048-  00  84  02  A6  04  20  5D  10
1050-  A4  08  D0  D9  E6  00  D0  BD
1058-  E6  01  D0  B9  60  8A  29  C0
1060-  85  05  4A  4A  05  05  85  05
1068-  8A  85  06  0A  0A  0A  26  06
1070-  0A  26  06  0A  66  05  A5  06
1078-  29  1F  05  E6  85  06  60  A0
1080-  01  84  04  84  03  88  84  02
1088-  A5  E6  09  04  85  06  84  05
1090-  B1  05  D0  02  09  80  A2  01
1098-  86  08  85  07  A4  02  A6  03
10A0-  E8  E8  E0  BF  90  0F  C8  C0
10A8-  28  90  06  C6  04  30  1B  A0
10B0-  00  84  02  A6  04  86  03  20
```

```
10B8- 5D 10 B1 05 D0 02 09 80
10C0- C5 07 D0 06 E6 08 D0 D4
10C8- C6 08 48 A0 00 A6 08 F0
10D0- 0E E0 04 B0 0A A5 07 20
10D8- F2 10 CA D0 FA F0 0D 98
10E0- 20 F2 10 8A 20 F2 10 A5
10E8- 07 20 F2 10 68 24 04 10
10F0- A5 60 91 00 E6 00 D0 02
10F8- E6 01 60
```

显示程序GP2C06清单。

```
10  POKE   16304,0
20  POKE  - 16297,0
30  POKE  - 16302,0
40  POKE  - 16300,0
50  END
```

如果要看一下压缩图形是什么样子，可在] BLOAD TC1↙之前执行HGR命令，然后在监控状态下键入2000<6000.7FFFM↙，再运行显示程序GP2C06即可。

对于连续存贮多页压缩图形资料，原则上按照上述图形压缩的步骤，依次处理，但要注意重新存贮的地址不能重叠，应严格分开。同理，还原多页压缩图形资料，也应注意各个被压缩的图形重新存放的地址，否则还原有误。

第五节　压缩存贮实例

程序P30-1可以完成5幅画面的联动显示。其中CC1中存了两幅画面，CC2中存了3幅画面。例如有两个图型C1，C2，它们是两幅不同画面的图形资料，按下述步骤将它们重存在一个程序CC1中：

· BLOAD C1，A$2000↙

· BLOAD C2, A$4000↙

· BSAVE CC1, A$2000, L$5FFF↙

同理，另外3幅画面C3, C4, C5, 也可以存在同一个程序CC2中，即：

· BLOAD C3, A$2000↙

· BLOAD C4, A$4000↙

· BLOAD C5, A$6000↙

· BSAVE CC2, A$2000, L$7FFF↙,

程序RAM1, 其功能是将调进内存中的CC1图形资料，转存在语言卡（即16K RAM CARD）中，但必须在BLOAD CC1, A$2000↙及BLOAD RAM1, A$8300↙后，执行CALL 33555。

程序P30，能完成5幅画面的联动显示，这也要求在BLOAD CC2, A$2000↙及BLOAD, P30, A$8000↙后，执行CALL32964。

下面给出3个程序清单。

程序P30-1（BASIC语言，10—80句）：

```
10 D$ =  CHR$ (4)
20  PRINT D$"BLOADCC1,A$2000"
30  PRINT D$"BLOADRAM1,A$8300"
40  CALL 33555
50  PRINT D$"BLOADCC2,A$2000"
60  PRINT D$"BLOADP30,A$8000"
70  CALL 32964
80  TEXT : HOME : END
```

程序RAM1（机器语言，$8300—$836B）：

```
8300-  A9 06 8D 39 83 8D 58 83
8308-  A9 08 8D 3B 83 8D 5A 83
8310-  4C 23 83 A9 06 8D 3B 83
8318-  8D 5A 83 A9 08 8D 39 83
8320-  8D 58 83 AE 83 C0 AE 83
8328-  C0 A9 D0 85 07 A9 20 85
8330-  09 A9 00 85 06 85 08 A8
8338-  B1 06 91 08 88 D0 F9 E6
8340-  07 E6 09 A5 09 C9 50 D0
8348-  EF AE 8B C0 AE 8B C0 A9
8350-  D0 85 07 A9 50 85 09 B1
8358-  06 91 08 88 D0 F9 E6 07
8360-  E6 09 A5 09 C9 60 D0 EF
8368-  AD 81 C0 60
```

程序P 30（机器语言，＄8000—＄80 F7）：

```
8000-  A9 00 85 1E A9 00 95 1F
8008-  A2 00 BD 00 81 85 07 BD
8010-  00 82 85 06 E8 BD 00 81
8018-  85 09 BD 00 82 85 08 E0
8020-  01 D0 08 A0 27 B1 06 48
8028-  88 10 FA A0 27 B1 08 91
8030-  06 88 10 F9 E0 BF D0 D2
8038-  A5 1E C9 00 D0 08 A9 20
8040-  8D 8F 80 4C 6F 80 C9 01
8048-  D0 08 A9 40 8D 8F 80 4C
8050-  6F 80 C9 02 D0 0E AD 83
8058-  C0 AD 83 C0 A9 B0 8D 8F
8060-  80 4C 6F 80 AD 8B C0 AD
8068-  8B C0 A9 D0 8D 8F 80 A5
8070-  08 85 06 A5 09 85 07 A6
8078-  1F BD 00 81 A4 1E C0 03
8080-  D0 0B C9 30 90 07 48 A9
8088-  A0 8D 8F 80 68 18 69 A0
8090-  85 09 BD 00 82 85 08 A0
8098-  27 B1 08 91 06 88 10 F9
```

```
80A0- A0 00 68 91 08 C8 C0 28
80A8- 90 F8 E6 1F A5 1F C9 C0
80B0- D0 0F E6 1E A5 1E C9 04
80B8- D0 04 AD 81 C0 60 4C 04
80C0- 80 4C 08 80 A9 20 85 E6
80C8- A2 00 8A 48 20 11 F4 68
80D0- AA A5 26 9D 00 82 A5 27
80D8- 9D 00 81 E8 E0 C0 D0 EA
80E0- AD 54 C0 AD 50 C0 AD 57
80E8- C0 AD 53 C0 AD 10 C0 20
80F0- 00 80 AD 00 C0 10 F5 60
```

5幅画面的联动显示，操作方法十分简单，即：RUN P30-1↙即可。但应注意CC1，CC2，RAM1，P30 4个程序都应存于同一盘片中。

第 三 篇

机器语言绘图技术

随着微型计算机的广泛应用和发展，微电脑绘图已逐步成为涉及无线电通讯、机械、建筑、教育等多种行业的一种重要设计技术。作为一种设计手段，它使各领域的研究者们能够将大脑中虚构的模型准确地反映出来，并通过对这种实体模型的不断改进设计出更为完美的装置。

但是，在许多程序设计人员看来，绘图似乎比计算机的其它功能更难更高深一些，因而应用并不广泛。尤其是除了BASIC、FORTRAN等高级语言之外，而触及到机器语言或汇编语言的情况更是如此。实际上，APPLE Ⅱ是一种比较好的绘图工具，而6502机器语言无论在图形效果还是在绘图速度上都比高级语言强得多。比如，使用机器语言可以设计出一个比较完美的游戏程序，高级语言就很困难。

本篇的内容就是介绍6502机器语言在绘图技术中的应用。

第一章　机器语言绘图

第一节　APPLE Ⅱ的图形空间

APPLE Ⅱ的屏幕显示采用记忆体映象输出的方式，也即其屏幕显示是通过对随机存取存贮器特定区域的扫描，将存贮器中的内容转换成象素而实现的。图形方式也一样，当这些象素按照指定位置排列在屏幕上时就构成了图形。因此，它在RAM之中有硬性设置的绘图空间，这些空间随绘图模式和页面的不同而不同。在APPLE Ⅱ的绘图系统中有两种模式，即低分辨和高分辨模式，每一种模式分别有两个页面，其在RAM中的空间分配如下：

第一低分辨作图页：$ 400—$ 7FF（1024—2047），1K；

第二低分辨作图页：$ 800—$ BFF（2048—3071），1K；

第一高分辨作图页：$ 2000—$ 3FFF（8192—16383），8K；

第二高分辨作图页：$ 4000—$ 5FFF（16384—24575），8K。

低分辨时水平为40点，垂直为48点；高分辨时水平为280点，垂直为192点。

第二节　绘图功能的调用

在APPLESOFT中我们使用GR、HGR、PLOT、HPLOT等语句来进行图形绘画，那么在机器语言中又如何实现这些功能呢？

在存贮器的ROM中，有实现各种功能的子程序，其中一部分就是用来进行绘图的。它包括启动作图页面、设置颜色、画点画线等等。当设置好所需要的参数，调用相应的子程序时，就能完成一定的绘图功能。

一、*启动作图页面*

APPLESOFT中的GR、HGR、HGR2分别是启动低分辨作图页、第一高分辨作图页和第二高分辨作图页的语句，在机器语言中也可以很方便地实现这一功能，其指令是：

20　40　FB　JSR$FB40：启动低分辨作图页；

20　E2　F3　JSR$F3E2：启动第一高分辨作图页；

20　D8　F3　JSR$F3D8：启动第二高分辨作图页。

这三个指令没有调用参数，执行时相应页面清为黑色。当然，在机器语言中也可以使用软体开关，方法与APPLESOFT类似，见表3.1

如用下列指令就可启动第二高分辨作图页而不清除原画面：

8D　50　C0　STA$C050　设定图形状态；

8D　52　C0　STA$C052　设定全屏幕方式；

8D　55　C0　STA$C055　设定第二页面；

8D　57　C0　STA$C057　设定高分辨模式。

表 3.1

屏幕状态	机器语言指令
图形状态	8D 50 C0 STA$C050
文本状态	8D 51 C0 STA$C051
全屏幕方式	8D 52 C0 STA$C052
混合方式	8D 53 C0 STA$C053
第一页面	8D 54 C0 STA$C054
第二页面	8D 55 C0 STA$C055
低分辨率	8D 56 C0 STA$C056
高分辨率	8D 57 C0 STA$C057

二、在屏幕上画点

画点是屏幕作图的最基本操作，根据作图模式的不同，可以在机器语言中实现。

• 低分辨率画点

低分辨率画点的子程序起始地址是$F800，点的颜色代码($00—$0F)存于$30单元中，纵坐标和横坐标分别存于累加器A和暂存器Y中。

例1，见程序GP3101。

```
GP3101

0300-    20 58 FC     JSR    $FC58
0303-    20 40 FB     JSR    $FB40
0306-    A9 FF        LDA    #$FF
0308-    85 30        STA    $30
030A-    A0 00        LDY    #$00
030C-    A9 00        LDA    #$00
030E-    20 00 F8     JSR    $F800
0311-    60           RTS
```

程序说明：

$0300：清除当前屏幕为黑。

$0303：启动低分辩作图页。

$0306—$0309：颜色代码存入$30中。

$030A—$030D：设置横坐标为$00，纵坐标为$00。

$030E：调用画点子程序。

在监控状态下输入：

300G↙

则屏幕左上角画出一个点。

· 高分辨率画点

高分辨率画点的子程序起始地址是$F457，点的纵坐标在累加器A中，横坐标因最大为279，大于十六进制的$FF，所以分高位和低位部分，高位在暂存器Y中，低位在暂存器X中。

另外，高分辨率画点的颜色设置必须通过调用$F6F0开始的子程序来实现，调用前将颜色代码（$00—$07）存入暂存器X中。

例2，见程序GP3102。

```
GP3102

0300-    20 D8 F3     JSR    $F3D8
0303-    A2 03        LDX    #$03
0305-    20 F0 F6     JSR    $F6F0
0308-    A9 00        LDA    #$00
030A-    A2 17        LDX    #$17
030C-    A0 01        LDY    #$01
030E-    20 57 F4     JSR    $F457
0311-    60           RTS
```

程序说明：

$0300：启动第二高分辨作图页。

$0303：取得颜色代码。

$0305：设置绘图颜色。

$0308：设置纵坐标为$00。

$030A—$030D：设置横坐标低位为$17，高位为$01。

$030E：调用画点子程序。

键入：300G↙后，屏幕右上角(279，0)处画出一个点。

三、在屏幕上画线

知道了画点的方法，实际上就可以进行画线了，这只要将一条线段上所有点的坐标参量设置好，连续调用画点子程序。但这样做无疑比较麻烦。根据绘图模式的不同，在系统的ROM区中已经设计好相应的画线程序，调用这些程序就能较方便地画出一条线段来。

• 低分辨率画线

低分辨率画线程序有水平线和垂直线两种，画水平线的子程序起始地址是$F819，其颜色代码在$30中，纵坐标在累加器A中，横坐标起始值在暂存器Y中，结束值在$2C中。

例3，见程序GP3103。

```
GP3103

0300-   20 58 FC    JSR   $FC58
0303-   20 40 FB    JSR   $FB40
0306-   A9 FF       LDA   #$FF
0308-   85 30       STA   $30
030A-   A9 27       LDA   #$27
030C-   85 2C       STA   $2C
030E-   A9 00       LDA   #$00
0310-   A0 00       LDY   #$00
```

```
0312-    20 19 F8     JSR    $F819
0315-    60           RTS
```

程序说明：

$0300：清除当前屏幕为黑。

$0303：启动低分辨作图页。

$0306—$0309：颜色代码存入$30中。

$030A—$030D：置横坐标结束值。

$030E：置纵坐标为$00。

$0310：置横坐标起始值为$00。

$0312：调用画线子程序。

程序GP3103运行后将在屏幕的上方画一根从左到右的水平线。

画垂直线的子程序起始地址是$F828，其颜色代码在$30中，横坐标在暂存器Y中，纵坐标起始值在累加器A中，结束值在$2D中。

例4，见程序GP3104。

```
     GP3104
0300-    20 58 FC     JSR    $FC58
0303-    20 40 FB     JSR    $FB40
0306-    A9 FF        LDA    #$FF
0308-    85 30        STA    $30
030A-    A9 10        LDA    #$10
030C-    85 2D        STA    $2D
030E-    A9 00        LDA    #$00
0310-    A0 05        LDY    #$05
0312-    20 28 F8     JSR    $F828
0315-    60           RTS
```

程序说明：

$0300：清除当前屏幕。

$0303：启动低分辨作图页面。

$0306—$0309：颜色代码存入$30中。

$030A—$030F：置纵坐标起始值为$00，结束值为$10。

$0310：置横坐标为$05。

$0312：调用画线子程序。

程序GP3104运行后将在屏幕左端X＝5的位置上画一垂直线。

·高分辨率画线

高分辨率画线是通过连结两个点来实现的，因此必须先设置起始点坐标，调用高分辨率画点程序（$F457）画出这个点，再定义结束点坐标，调用$F53A开始的画线子程序连结两个点之间的线段。位于$F53A的子程序将纵坐标存于暂存器Y中，横坐标低位存于累加器A中，高位存于暂存器X中。

例5，GP3105。

```
      GP3105

0300-    20 D8 F3      JSR    $F3D8
0303-    A2 03         LDX    #$03
0305-    20 F0 F6      JSR    $F6F0
0308     A0 00         LDY    #$00
030A-    A2 00         LDX    #$00
030C-    A9 00         LDA    #$00
030E-    20 57 F4      JSR    $F457
0311-    A0 BF         LDY    #$BF
0313-    A2 01         LDX    #$01
0315-    A9 17         LDA    #$17
0317-    20 3A F5      JSR    $F53A
031A-    60            RTS
```

程序说明：

$0300：启动第二高分辨作图页。

$0303：取得颜色代码。

$0305：设置颜色。

$0308：设置横坐标高位为$00。

$030A：设置横坐标低位为$00。

$030C：设置纵坐标为$00。

$030E：调用画点子程序。

$0311：设置纵坐标为$BF。

$0313：设置横坐标高位为$01。

$0315：设置横坐标低位为$17。

$0317：调用画线子程序。

$031A：返回。

程序GP3105设置好绘图模式和颜色后，先画出起始点($308—$310)，然后设置结束点，并调用画线子程序($311—$31A)。运行后，从屏幕左上角到右下角画出一线段。

四、高分辨率造型表操作

本篇第二部分将介绍造型表的设计和使用，为此我们先说明一下有关造型表操作的几个机器语言子程序。

· 调用造型的指针设置

子程序起始地址是$F730，调用前将造型序号存入暂存器X中。该子程序的功能主要是根据造型表存贮地址及造型序号取得造型向量的绝对存贮地址，并将地址的高位和低位分别存入零页单元$1B、$1A中，指令为：

```
A9  01     LDA#$01      造型序号为$01
20  30  F7 JSR$F730     调用子程序
```

· 造型显示位置的设置

子程序起始地址是＄F411，调用前将显示位置的纵坐标存入累加器A，横坐标低位存入暂存器X，高位存入暂存器Y中。该子程序的功能是将设置的显示位置(指造型起始点的显示位置)加以转换，并存入零页单元＄26、＄27、＄30和＄E5。其中＄26、＄27指向屏幕扫描行最左端所对应的地址，＄E5为指定地址相对于扫描行左端的地址，其值为＄00—＄27。

· 造型显示

使用指定颜色显示造型的子程序起始地址是＄F601，使用补色显示造型的子程序起始地址是＄F65D。调用前先将造型旋转值存入累加器A，造型向量绝对地址的低、高位元分别存入暂存器X和Y中。指令为：

```
A9 01     LDA#$01      造型旋转值为$01
A6 1A     LDX $1A      低位存入X
A4 1B     LDY $1B      高位存入Y
20 01 F6  JSR $F601    调用子程序
```

以上介绍的3个造型操作子程序实际上相当于APPLESOFT中的DRAW A AT X,Y和XDRAW A AT X,Y指令。这里A为造型在造型表中的序号，X及Y分别是造型起始点显示位置的横坐标和纵坐标。

除此之外，还有一个很重要的参量，这就是造型的放大值，这个值必须在调用造型显示子程序之前置入零页单元＄E7中。

至此，我们已将机器语言绘图的主要部分介绍完了。使用这些子程序就可以在6502机器语言中实现很多绘图功能。下一部分首先简单介绍一下造型和造型表的结构，然后逐步完善一个造型设计和使用的应用软件。

第二章　造型的设计和使用

第一节　造型原理及造型表结构

造型是APPLE-Ⅱ提供的一种强有力的绘图手段，利用造型可以设计出各种简单和复杂的图案，这些图案可以在高分辨率作图页面的任一指定位置上显示，并能通过一定的指令对图形进行旋转、放大处理。因而在APPLEⅡ绘图中得到广泛应用。

另外，造型图案光滑、细腻，显示速度快，这也是它不同于其它方式绘图的特点。

那么造型是怎样得到的呢？我们知道，作图的过程实际上就是光点在屏幕上移动并留下痕迹的过程，这个过程的每一步都是光点从一个位置移到另一个位置，把这种带方向性的移动过程用矢量形式记录下来，就能够得到图形，这就是造型。

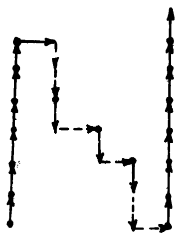
根据其移动方向和是否画点的情况把矢量分解成8种，各以一定的代码代替，见图3.1。

当得到一连串的二进制代码后，按照一定的规则将之转换成十六进制码，这就是造型元素，见图3.2。

图3.2右边的12个数就是字符“N”的造型，其中最后的00字节是造型结束的标志。把这种造型按照一定的结构组织起来就成了造型表。

矢　量	作　　用	二进制代码
↑	上移而不画点	000
→	右移而不画点	001
↓	下移而不画点	010
←	左移而不画点	011
↑	上移并画点	100
→	右移并画点	101
↓	下移并画点	110
←	左移并画点	111

图 3.1



00100100		24	
00100100		24	
00100100		24	
10010101		95	
00001110	⇒	0E	
00001110		0E	
01010110		56	
00100100		24	
00100100		24	
00100100		24	
00000100		04	
00000000		00	

图 3.2

第二节　造型表的构成

造型表由两个部分组成，即造型目录和造型向量。

造型表的第一个字节存放造型的数目，其值为＄00—＄FF，即一个造型表最多可存放255个造型；第二个字节不用；从第三个字节开始，每两个字节一组存放各造型相对于造型表首的存放地址：头一个字节为地址低位，后一个字节为地址高位。

第三节　造型表的使用

当建立好一个造型表后，就可以使用这个表中的任一造型来进行图案设计。使用前先设置零页指针＄E8、＄E9(即十进制232、233)，使其指向表的起始位置，如造型表起始位置为＄6000(24576)，则＄E8、＄E9中的值应分别是＄00和＄60。然后启动高分辨作图页，调用BASIC指令或相应的机器语言子程序来绘图。

前面已经介绍过，造型的显示是通过调用DRAW（＄F601)或XDRAW（＄F6D5）子程序来实现的，两者的区别是：DRAW以指定颜色显示造型，而XDRAW则以屏幕底色的补色来显示造型。因而当用DRAW在指定位置上显示一个造型后，再用XDRAW在同一位置上显示同一造型，正好起到清除原造型的作用。

另外，造型的旋转角度和放大倍数是DRAW和XDRAW子程序调用前就必须设置好的两个参量。

放大倍数在＄00—＄FF之间（放大倍数为＄00时实际值为＄100，即256倍)，由于屏幕范围的限制，一般造型的放

大倍数不可能很大，否则就会超出屏幕而发生变形。

旋转角度在＄00—＄3F之间，但实际造型能够旋转的角度还决定于放大倍数。当放大倍数：

为 1 时，能旋转的角度为0(0°)、16(90°)、32(180°)、48(270°)；

为 2 时，能旋转的角度为0(0°)、8(45°)、16(90°)、24(135°)、32(180°)、40(225°)、48(270°)、56(315°)。

大于或等于 5 时，能旋转64个角度。

通过上面的介绍，可以看到使用造型来绘图有许多优点，但造型的建立和造型表的组合却比较麻烦和费时。为此，本篇的下一部分将逐步完成一个造型表设计的机器语言程序，这一程序具有造型表的调入、存贮、初始化、造型建立、内插、删除以及图形设计等功能。从中，读者可以看到机器语言造型设计的大概情况。

第三章　造型程序设计

本章所提供的这一造型设计程序全部使用6502机器语言编成。按照各程序块的功能不同，我们将其分为8个部分分别介绍，最后介绍程序的操作使用。

第一节　参数设置

见程序GP 3301清单。

GP3301

```
0801-   A9 4C       LDA   #$4C
0803-   8D F5 03    STA   $03F5
0806-   A9 20       LDA   #$20
0808-   8D F6 03    STA   $03F6
080B-   A9 08       LDA   #$08
080D-   8D F7 03    STA   $03F7
0810-   A2 00       LDX   #$00
0812-   BD 00 13    LDA   $1300,X
0815-   9D 00 02    STA   $0200,X
0818-   E8          INX
0819-   E0 0D       CPX   #$0D
081B-   D0 F5       BNE   $0812
081D-   20 CD 9F    JSR   $9FCD
0820-   A9 00       LDA   #$00
0822-   8D 01 60    STA   $6001
0825-   A9 00       LDA   #$00
0827-   A2 0B       LDX   #$0B
0829-   9D F4 00    STA   $00F4,X
```

```
082C-    CA              DEX
082D-    D0 FA           BNE    $0829
082F-    85 F2           STA    $F2
0831-    A9 01           LDA    #$01
0833-    8D 0B 03        STA    $030B
0836-    A9 00           LDA    #$00
0838-    85 E8           STA    $E8
083A-    A9 03           LDA    #$03
083C-    85 E9           STA    $E9
083E-    A9 00           LDA    #$00
0840-    8D 50 C0        STA    $C050
0843-    8D 52 C0        STA    $C052
0846-    8D 54 C0        STA    $C054
0849-    8D 57 C0        STA    $C057
084C-    A2 00           LDX    #$00
084E-    BD 1C 13        LDA    $131C,X
0851-    9D 00 03        STA    $0300,X
0854-    E8              INX
0855-    E0 0B           CPX    #$0B
0857-    D0 F5           BNE    $084E
0859-    20 0C FD        JSR    $FD0C
```

程序说明：

$0801—$080F：设置&入口为$0820。

$0810—$081F：调入第一高分辨图形SHAP0。

$0820—$0835：初始化有关存贮单元。

$0836—$083D：设置光标造型始址为$0300。

$083E—$084B：启动第一高分辨作图页。

$084C—$0858：将光标造型移入$0300。

程序GP3301首先设置&入口，以便退出或中断运行后能随时从键盘输入“&”指令重新启动。然后调入一个名为SHAPE0的二进制文件，这一文件实际上是一幅第一高分辨作图页上的图形信息，它使用造型汉字显示程序的各个

功能，其格式：

——操作指令——
[1] 造型表调入
[2] 造型表存贮
[3] 造型表初置
[4] 造型建立
[5] 造　型
[3] 造　型
[7] 图形设计
——请选择?

图形信息调入内存后，对程序中要用到的一些指针和参量存贮单元进行初值化，并启动高分辨作图第一页面，以显示其操作指令。最后将光点造型移入$300开始的地址中，等待操作者键入功能序号。

第二节　造型表的调入和存贮

见程序GP 3302清单。

```
    GP3302
085C-    C9 B1        CMP    #$B1
085E-    D0 3B        BNE    $089B
0860-    F0 17        BEQ    $0879
0862-    20 58 FC     JSR    $FC58
0865-    20 2F FB     JSR    $FB2F
0868-    A2 00        LDX    #$00
086A-    BD 0C 13     LDA    $130C,X
086D-    20 ED FD     JSR    $FDED
0870-    E8           INX
0871-    E0 0F        CPX    #$0F
0873-    D0 F5        BNE    $086A
```

```
0875-   20 6F FD    JSR   $FD6F
0878-   60          RTS
0879-   20 62 08    JSR   $0862
087C-   E8          INX
087D-   CA          DEX
087E-   BD 00 02    LDA   $0200,X
0881-   9D 05 02    STA   $0205,X
0884-   E0 00       CPX   #$00
0886-   D0 F5       BNE   $087D
0888-   A2 05       LDX   #$05
088A-   BD FF 12    LDA   $12FF,X
088D-   9D FF 01    STA   $01FF,X
0890-   CA          DEX
0891-   D0 F7       BNE   $088A
0893-   48          PHA
0894-   48          PHA
0895-   20 CD 9F    JSR   $9FCD
0898-   4C 25 08    JMP   $0825
089B-   C9 B2       CMP   #$B2
089D-   F0 03       BEQ   $08A2
089F-   4C 21 09    JMP   $0921
08A2-   20 62 08    JSR   $0862
08A5-   A9 AC       LDA   #$AC
08A7-   9D 00 02    STA   $0200,X
08AA-   E8          INX
08AB-   A9 C1       LDA   #$C1
08AD-   9D 00 02    STA   $0200,X
08B0-   E8          INX
08B1-   A9 A4       LDA   #$A4
08B3-   9D 00 02    STA   $0200,X
08B6-   E8          INX
08B7-   A9 B6       LDA   #$B6
08B9-   9D 00 02    STA   $0200,X
08BC-   E8          INX
08BD-   A9 B0       LDA   #$B0
08BF-   9D 00 02    STA   $0200,X
08C2-   E8          INX
```

```
08C3-   9D 00 02    STA   $0200,X
08C6-   E8          INX
08C7-   9D 00 02    STA   $0200,X
08CA-   E8          INX
08CB-   A9 AC       LDA   #$AC
08CD-   9D 00 02    STA   $0200,X
08D0-   E8          INX
08D1-   A9 CC       LDA   #$CC
08D3-   9D 00 02    STA   $0200,X
08D6-   E8          INX
08D7-   8A          TXA
08D8-   48          PHA
08D9-   A5 FD       LDA   $FD
08DB-   38          SEC
08DC-   E9 60       SBC   #$60
08DE-   85 9E       STA   $9E
08E0-   A6 FC       LDX   $FC
08E2-   86 9F       STX   $9F
08E4-   A2 90       LDX   #$90
08E6-   38          SEC
08E7-   20 A0 EB    JSR   $EBA0
08EA-   20 34 ED    JSR   $ED34
08ED-   68          PLA
08EE-   AA          TAX
08EF-   A0 00       LDY   #$00
08F1-   B9 00 01    LDA   $0100,Y
08F4-   F0 0A       BEQ   $0900
08F6-   09 80       ORA   #$80
08F8-   9D 00 02    STA   $0200,X
08FB-   E8          INX
08FC-   C8          INY
08FD-   4C F1 08    JMP   $08F1
0900-   A9 8D       LDA   #$8D
0902-   9D 00 02    STA   $0200,X
0905-   E8          INX
0906-   CA          DEX
0907-   BD 00 02    LDA   $0200,X
090A-   9D 05 02    STA   $0205,X
```

```
090D-    E0 00       CPX   #$00
090F-    D0 F5       BNE   $0906
0911-    A2 00       LDX   #$00
0913-    BD 3X 13    LDA   $133C,X
0916-    9D 00 02    STA   $0200,X
0919-    E8          INX
091A-    E0 05       CPX   #$05
091C-    D0 F5       BNE   $0913
091E-    4C 93 08    JMP   $0893
```

程序说明：

$ 0862—$ 0878：输入文件名。

$ 087C —$ 0894：设置BLOAD指令。

$ 0895—$ 0897：调入造型表。

$ 08A5 —$ 091D：设置BSAVE指令。

$ 091E —$ 0920：存贮造型表。

当操作者从键盘按入"1"或"2"后（不必按回车），即转为执行造型表的调入或存贮功能。

首先要求输入造型表文件名，再调入或存贮造型表文件。调入和存贮过程实际上是执行BLOAD〈文件名〉和BSAVE〈文件名〉，A$6000，L〈长度〉命令。

第三节　造型表初置

见程序GP 3303清单。

```
0921-    C9 B3       CMP   #$B3
0923-    F0 03       BEQ   $0928
0925-    4C B7 09    JMP   $09B7
0928-    20 58 FC    JSR   $FC58
092B-    20 2F FB    JSR   $FB2F
092E-    A2 00       LDX   #$00
0930-    BD 41 13    LDA   $1341,X
0933-    20 ED FD    JSR   $FDED
0936-    E8          INX
```

```
0937-   E0 1B       CPX   #$1B
0939-   D0 F5       BNE   $0930
093B-   20 6F FD    JSR   $FD6F
093E-   A9 00       LDA   #$00
0940-   85 B8       STA   $B8
0942-   A9 02       LDA   #$02
0944-   85 B9       STA   $B9
0946-   A2 00       LDX   #$00
0948-   BD 00 02    LDA   $0200,X
094B-   C9 8D       CMP   #$8D
094D-   F0 09       BEQ   $0958
094F-   29 7F       AND   #$7F
0951-   9D 00 02    STA   $0200,X
0954-   E8          INX
0955-   4C 48 09    JMP   $0948
0958-   20 67 DD    JSR   $DD67
095B-   20 52 E7    JSR   $E752
095E-   20 8E FD    JSR   $FD8E
0961-   A5 51       LDA   $51
0963-   F0 03       BEQ   $0968
0965-   4C 2E 09    JMP   $092E
0968-   A5 50       LDA   $50
096A-   8D 00 60    STA   $6000
096D-   A2 01       LDX   #$01
096F-   A9 00       LDA   #$00
0971-   9D 00 60    STA   $6000,X
0974-   E8          INX
0975-   D0 FA       BNE   $0971
0977-   9D 00 61    STA   $6100,X
097A-   E8          INX
097B-   D0 FA       BNE   $0977
097D-   AD 00 60    LDA   $6000
0980-   18          CLC
0981-   2A          ROL
0982-   85 FC       STA   $FC
0984-   A9 00       LDA   #$00
0986-   69 00       ADC   #$00
0988-   85 FD       STA   $FD
098A-   A9 02       LDA   #$02
098C-   18          CLC
098D-   65 FC       ADC   $FC
098F-   85 FC       STA   $FC
0991-   A9 60       LDA   #$60
0993-   65 FD       ADC   $FD
0995-   85 FD       STA   $FD
0997-   AD 00 60    LDA   $6000
099A-   18          CLC
```

```
099B-    2A           ROL
099C-    8D 02 60     STA    $6002
099F-    A9 00        LDA    #$00
09A1-    69 00        ADC    #$00
09A3-    8D 03 60     STA    $6003
09A6-    AD 02 60     LDA    $6002
09A9-    18           CLC
09AA-    69 02        ADC    #$02
09AC-    8D 02 60     STA    $6002
09AF-    90 03        BCC    $09B4
09B1-    EE 03 60     INC    $6003
09B4-    4C 25 08     JMP    $0825
```

程序说明：

$0928—$093D：要求输入造型数目。

$093E—$0960：转换成十六进制。

$0961—$0967：如大于$FF则转为重新输入。

$0968—$096C：置入造型表第一个单元。

$096D—$09B3：造型表目录区初始化。

由于造型表的结构主要决定于造型的数目，因此造型表初置程序首先要求输入造型的数目。然后将这一数目转换十六进制并置入表的第一个单元，再对表的目录索引区进行初始化。

在本软件中，利用造型表的第二个单元$6001来存贮当前造型的实际数目。因此一开始$6001清为零。

第四节 建立造型

建立造型是本软件的主要功能，它首先在屏幕上绘制出一个25×15的表格，通过操作者键入的指令控制绘图光标的移动，并将移动所得的矢量经转换置入造型表的特定位置。同时在屏幕下方开辟一个窗口，用来显示当前造型的情况。

建立造型的指令有绘图光标的上、下、左、右移动、绘

点、清除、结束等（详见操作使用部分）。

下面我们将建立造型部分，分几段进行介绍。

一、造型存放地址的确定

见程序GP 3304清单。

```
GP3304
09B7-    C9 B4       CMP   #$B4
09B9-    F0 03       BEQ   $09BE
09BB-    4C 89 0F    JMP   $0F89
09BE-    AD 01 60    LDA   $6001
09C1-    CD 00 60    CMP   $6000
09C4-    D0 19       BNE   $09DF
09C6-    20 58 FC    JSR   $FC58
09C9-    20 2F FB    JSR   $FB2F
09CC-    A2 00       LDX   #$00
09CE-    BD 27 13    LDA   $1327,X
09D1-    20 ED FD    JSR   $FDED
09D4-    E8          INX
09D5-    E0 15       CPX   #$15
09D7-    D0 F5       BNE   $09CE
09D9-    20 0C FD    JSR   $FD0C
09DC-    4C 25 08    JMP   $0825
09DF-    A9 02       LDA   #$02
09E1-    85 B8       STA   $B8
09E3-    A9 60       LDA   #$60
09E5-    85 B9       STA   $B9
09E7-    AD 01 60    LDA   $6001
03EA-    18          CLC
09EB-    2A          ROL
09EC-    90 02       BCC   $09F0
09EE-    E6 B9       INC   $B9
09F0-    18          CLC
09F1-    65 B8       ADC   $B8
09F3-    85 B8       STA   $B8
09F5-    90 02       BCC   $09F9
09F7-    E6 B9       INC   $B9
09F9-    A0 00       LDY   #$00
```

```
09FB-     B1 B8        LDA   ($B8),Y
09FD-     48           PHA
09FE-     C8           INY
09FF-     B1 B8        STA   ($B8),Y
0A01-     85 B9        STA   $B9
0A03-     68           PLA
0A04-     85 B8        STA   $B8
0A06-     A9 60        LDA   #$60
0A08-     18           CLC
0A09-     65 B9        ADC   $B9
0A0B-     85 FD        STA   $FD
0A0D-     A5 B8        LDA   $B8
0A0F-     85 FC        STA   $FC
0A11-     AD 01 60     LDA   $6001
0A14-     D0 1F        BNE   $0A35
0A16-     AD 00 60     LDA   $6000
0A19-     18           CLC
0A1A-     2A           ROL
0A1B-     85 FC        STA   $FC
0A1D-     A9 00        LDA   #$00
0A1F-     69 00        ADC   #$00
0A21-     85 FD        STA   $FD
0A23-     A9 60        LDA   #$60
0A25-     18           CLC
0A26-     65 FD        ADC   $FD
0A28-     85 FD        STA   $FD
0A2A-     A5 FC        LDA   $FC
0A2C-     18           CLC
0A2D-     69 02        ADC   #$02
0A2F-     85 FC        STA   $FC
0A31-     90 02        BCC   $0A35
0A33-     E6 FD        INC   $FD
0A35-     AD 0B 03     LDA   $030B
0A38-     EA           NOP
0A39-     EA           NOP
0A3A-     EE 01 60     INC   $6001
```

程序说明：

$09BE－$09C3：当前造型表中造型是否已满。
$09C6－$09DE：如是则显示造型表已满，返回。
$09DF－$0A34：计算造型存放地址

建立造型一开始首先判断造型表是否已满。因为在造型表初置时已经规定了造型表可容纳造型的最大数目，因此只要比较当前表中的造型数目与最大数目即可判断是否存满。如已存满，就不能再建立造型而返回原清单。

如尚未存满则计算该造型的存放地址。存放地址的计算分两种情况：

当表中尚没有造型时，当前造型（即第一个造型）的始址在目录区后的第一个单元：最大造型数＊2＋2处，实际地址是：最大造型数＊2＋$6002；

当表中已有建立好的造型时，由于前一个造型结束时已算出下一个造型存放始址的偏移量并置入特定单元，因此将这一偏移量＋$6000就是实际地址。

计算出的存放地址高、低位分别存在零页单元$FD、$FC中。

二、表格绘制及初始位置确定

见程序GP3305清单。

```
      GP3305

0A3D-    20 D8 F3    JSR    $F3D8
0A40-    A2 03       LDX    #$03
0A42-    20 F0 F6    JSR    $F6F0
0A45-    A9 01       LDA    #$01
0A47-    85 E7       STA    $E7
0A49-    A9 00       LDA    #$00
0A4B-    8D 0C 03    STA    $030C
0A4E-    A9 00       LDA    #$00
```

```
0A50-   85 00      STA  $00
0A52-   A9 00      LDA  #$00
0A54-   A6 00      LDX  $00
0A56-   A0 00      LDY  #$00
0A58-   20 57 F4   JSR  $F457
0A5B-   A0 96      LDY  #$96
0A5D-   A5 00      LDA  $00
0A5F-   A2 00      LDX  #$00
0A61-   20 3A F5   JSR  $F53A
0A64-   A5 00      LDA  $00
0A66-   18         CLC
0A67-   69 0A      ADC  #$0A
0A69-   85 00      STA  $00
0A6B-   C9 0A      CMP  #$0A
0A6D-   B0 E3      BCS  $0A52
0A6F-   A9 00      LDA  #$00
0A71-   85 00      STA  $00
0A73-   A5 00      LDA  $00
0A75-   A2 00      LDX  #$00
0A77-   A0 00      LDY  #$00
0A79-   20 57 F4   JSR  $F457
0A7C-   A4 00      LDY  $00
0A7E-   A2 00      LDX  #$00
0A80-   A9 FA      LDA  #$FA
0A82-   20 3A F5   JSR  $F53A
0A85-   A5 00      LDA  $00
0A87-   18         CLC
0A88-   69 0A      ADC  #$0A
0A8A-   85 00      STA  $00
0A8C-   C9 A0      CMP  #$A0
0A8E-   D0 E3      BNE  $0A73
0A90-   20 58 FC   JSR  $FC58
0A93-   20 2F FB   JSR  $FB2F
0A96-   4C DA 0A   JMP  $0ADA
0A99-   A2 00      LDX  #$00
0A9B-   BD 5C 13   LDA  $135C,X
0A9E-   20 ED FD   JSR  $FDED
```

```
0AA1-    E8          INX
0AA2-    E0 10       CPX   #$10
0AA4-    D0 F5       BNE   $0A9B
0AA6-    98          TYA
0AA7-    20 ED FD    JSR   $FDED
0AAA-    A9 BF       LDA   #$BF
0AAC-    20 ED FD    JSR   $FDED
0AAF-    20 6F FD    JSR   $FD6F
0AB2-    A9 00       LDA   #$00
0AB4-    85 B8       STA   $B8
0AB6-    A9 02       LDA   #$02
0AB8-    85 B9       STA   $B9
0ABA-    A2 00       LDX   #$00
0ABC-    BD 00 02    LDA   $0200,X
0ABF-    C9 8D       CMP   #$8D
0AC1-    F0 09       BEQ   $0ACC
0AC3-    29 7F       AND   #$7F
0AC5-    9D 00 02    STA   $0200,X
0AC8-    E8          INX
0AC9-    4C BC 0A    JMP   $0ABC
0ACC-    20 67 DD    JSR   $DD67
0ACF-    20 52 E7    JSR   $E752
0AD2-    20 8E FD    JSR   $FD8E
0AD5-    A5 51       LDA   $51
0AD7-    D0 C0       BNE   $0A99
0AD9-    60          RTS
0ADA-    A0 D8       LDY   #$D8
0ADC-    20 99 0A    JSR   $0A99
0ADF-    A5 50       LDA   $50
0AE1-    C9 1A       CMP   #$1A
0AE3-    B0 F5       BCS   $0ADA
0AE5-    85 FE       STA   $FE
0AE7-    A0 D9       LDY   #$D9
0AE9-    20 99 0A    JSR   $0A99
0AEC-    A5 50       LDA   $50
0AEE-    C9 10       CMP   #$10
0AF0-    B0 F7       BCS   $0AE9
```

```
0AF2-      85 FF       STA    $FF
0AF4-      A2 00       LDX    #$00
0AF6-      A5 FE       LDA    $FE
0AF8-      18          CLC
0AF9-      65 FE       ADC    $FE
0AFB-      E8          INX
0AFC-      E0 09       CPX    #$09
0AFE-      D0 F8       BNE    $0AF8
0B00-      85 FE       STA    $FE
0B02-      A2 00       LDX    #$00
0B04-      A5 FF       LDA    $FF
0B06-      18          CLC
0B07-      65 FF       ADC    $FF
0B09-      E8          INX
0B0A-      E0 09       CPX    #$09
0B0C-      D0 F8       BNE    $0B06
0B0E-      85 FF       STA    $FF
0B10-      A2 01       LDX    #$01
0B12-      20 30 F7    JSR    $F730
0B15-      A5 FE       LDA    $FE
0B17-      F0 0E       BEQ    $0B27
0B19-      38          SEC
0B1A-      E9 05       SBC    #$05
0B1C-      85 FE       STA    $FE
0B1E-      A5 FF       LDA    $FF
0B20-      38          SEC
0B21-      E9 05       SBC    #$05
0B23-      85 FF       STA    $FF
```

程序说明：

$0A52—$0A8F：画出造型方框。

$0A99—$0AB1：输入起始位置坐标。

$0AB2—$0AD9：转换成十六进制。

$0ADA—$0AE6：取得横坐标X。

$0AE7—$0AF3：取得纵坐标Y。

$0AF4—$0B24：计算实际坐标位置。

前面一个程序段（$A3D—$A8F）所画的表格是一个25×15的表格，其实际位置是横向0—250，纵向0—150。绘图光标就在这一表格内移动（见图3.3）。

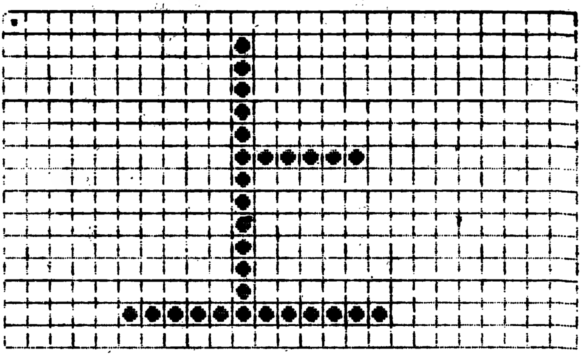


图 3.3

因此，接着要输入的起始位置坐标要求横坐标X在1—25之间，纵坐标Y在1—15之间。然后，将之转换成实际位置：

X＝10＊X－5

Y＝10＊Y－5

这一输入的起始位置就是造型的起点，也是造型旋转时的支点。

三、控制造型光标

见程序GP3306清单。

```
GP3306

0B25-        A0 00          LDY     #$00
0B27-        A6 FE          LDX     $FE
```

```
0B29-     20 11 F4    JSR   $F411
0B2C-     A6 1A       LDX   $1A
0B2E-     A4 1B       LDY   $1B
0B30-     A9 00       LDA   #$00
0B32-     20 01 F6    JSR   $F601
0B35-     A5 FE       LDA   $FE
0B37-     85 FA       STA   $FA
0B39-     A5 FF       LDA   $FF
0B3B-     85 FB       STA   $FB
0B3D-     A9 00       LDA   #$00
0B3F-     85 F6       STA   $F6
0B41-     A9 00       LDA   #$00
0B43-     A0 01       LDY   #$01
0B45-     91 FC       STA   ($FC),Y
0B47-     A5 E9       LDA   $E9
0B49-     48          PHA
0B4A-     A9 60       LDA   #$60
0B4C-     85 E9       STA   $E9
0B4E-     AE 01 60    LDX   $6001
0B51-     20 30 F7    JSR   $F730
0B54-     A9 AA       LDA   #$AA
0B56-     A0 00       LDY   #$00
0B58-     A2 80       LDX   #$80
0B5A-     20 11 F4    JSR   $F411
0B5D-     A9 00       LDA   #$00
0B5F-     A6 1A       LDX   $1A
0B61-     A4 1B       LDY   $1B
0B63-     20 01 F6    JSR   $F601
0B66-     68          PLA
0B67-     85 E9       STA   $E9
0B69-     20 0C FD    JSR   $FD0C
0B6C-     8D 0D 03    STA   $030D
0B6F-     C9 C9       CMP   #$C9
0B71-     D0 18       BNE   $0B8B
0B73-     A9 00       LDA   #$00
0B75-     85 F7       STA   $F7
```

```
0B77-    A5 FF       LDA   $FF
0B79-    C9 0A       CMP   #$0A
0B7B-    B0 06       BCS   $0B83
0B7D-    20 3A FF    JSR   $FF3A
0B80-    4C 41 0B    JMP   $0B41
0B83-    38          SEC
0B84-    E9 0A       SBC   #$0A
0B86-    85 FF       STA   $FF
0B88-    4C 2E 0C    JMP   $0C2E
0B8B-    C9 CB       CMP   #$CB
0B8D-    D0 15       BNE   $0BA4
0B8F-    A9 01       LDA   #$01
0B91-    85 F7       STA   $F7
0B93-    A5 FE       LDA   $FE
0B95-    C9 F1       CMP   #$F1
0B97-    90 03       BCC   $0B9C
0B99-    4C 7D 0B    JMP   $0B7D
0B9C-    18          CLC
0B9D-    69 0A       ADC   #$0A
0B9F-    85 FE       STA   $FE
0BA1-    4C 2E 0C    JMP   $0C2E
0BA4-    C9 CD       CMP   #$CD
0BA6-    D0 15       BNE   $0BBD
0BA8-    A9 02       LDA   #$02
0BAA-    85 F7       STA   $F7
0BAC-    A5 F‾       LDA   $FF
0BAE-    C9 8D       CMP   #$8D
 BB0-    90 03       BCC   $0BB5
0BB2-    4C 7D 0B    JMP   $0B7D
0BB5-    18          CLC
0BB6-    69 0A       ADC   #$0A
0BB8-    85 FF       STA   $FF
0BBA-    4C 2E 0C    JMP   $0C2E
0BBD-    C9 CA       CMP   #$CA
0BBF-    D0 15       BNE   $0BD6
0BC1-    A9 03       LDA   #$03
0BC3-    85 F7       STA   $F7
```

```
0BC5-        A5 FE          LDA    $FE
0BC7-        C9 0A          CMP    #$0A
0BC9-        B0 03          BCS    $0BCE
0BCB-        4C 7D 0B       JMP    $0B7D
0BCE-        38             SEC
0BCF-        E9 0A          SBC    #$0A
0BD1-        85 FE          STA    $FE
0BD3-        4C 2E 0C       JMP    $0C2E
```

程序说明：

$0B25—$0B34：　显示光标造型。

$0B41—$0B68：　显示当前建立的造型。

$0B69：　等待按键。

$0B6F—$0B8A：如按键为I，则纵坐标递减$0A，上移一格。

$0B8B—$0BA3：如按键为K，则横坐标递增$0A，右移一格。

$0BA4—$0BBC：如按键为M，则纵坐标递增$0A，下移一格。

$0BBD—$0BD5：如按键为J，则横坐标递减$0A，左移一格。

造型光标是指示当前屏幕位置的标志。这一标志通过按入I、M、J、K 4个键来控制上、下、左、右的移动，当按入其中一个键时，首先判断光标是否会越出画好的表格，如是则按键无效，发出一声“嘟”后返回。否则改变坐标值，以便相应移动光标。

四、点的清除

见程序GP3307清单。

```
0BD6-    C9 D0       CMP  #$D0
0BD8-    D0 0A       BNE  $0BE4
0BDA-    A9 01       LDA  #$01
0BDC-    85 F5       STA  $F5
0BDE-    20 26 0D    JSR  $0D26
0BE1-    4C DB 0D    JMP  $0DDB
0BE4-    C9 D1       CMP  #$D1
0BE6-    D0 03       BNE  $0BEB
0BE8-    4C 66 0D    JMP  $0D66
0BEB-    C9 C5       CMP  #$C5
0BED-    F0 27       BEQ  $0C16
0BEF-    20 58 FC    JSR  $FC58
0BF2-    20 2F FB    JSR  $FB2F
0BF5-    A2 00       LDX  #$00
0BF7-    BD 6D 13    LDA  $136D,X
0BFA-    20 ED FD    JSR  $FDED
0BFD-    E8          INX
0BFE-    E0 54       CPX  #$54
0C00-    D0 F5       BNE  $0BF7
0C02-    20 0C FD    JSR  $FD0C
0C05-    A9 00       LDA  #$00
0C07-    8D 50 C0    STA  $C050
0C0A-    8D 52 C0    STA  $C052
0C0D-    8D 55 C0    STA  $C055
0C10-    8D 57 C0    STA  $C057
0C13-    4C 41 0B    JMP  $0B41
0C16-    A2 00       LDX  #$00
0C18-    20 F0 F6    JSR  $F6F0
0C1B-    A9 00       LDA  #$00
0C1D-    85 F5       STA  $F5
0C1F-    20 26 0D    JSR  $0D26
0C22-    A5 F4       LDA  $F4
0C24-    85 F6       STA  $F6
0C26-    A2 03       LDX  #$03
0C28-    20 F0 F6    JSR  $F6F0
```

```
0C2B-       4C B8 0D    JMP   $0DB8
```

程序说明：

$0BD6—$0BE3：画点。

$0BE4—$0BEA：退出。

$0BEE—$0C04：如按入非指令键，则显示提示信息。

$0C05—$0C12：返回原作图页。

$0C16—$0C2A：清除光点。

当按入P键画了一个不应画的点后，可按E键清除。如果按入的键不是指令键，则启动文本状态显示下面的提示信息：

"MOVE PLOT WITH FOLLOWING KEYS:
I—UP, M—DOWN, J—LEFT, K—RIGHT
P—PLOT, E—ERASE, Q—QUIT"

待按下任意一个键后又返回造型建立的页面。

五、矢量组合

见程序GP3308清单。

```
GP3308

0C2E-       A5 F6       LDA   $F6
0C30-       C9 D0       CMP   #$D0
0C32-       D0 07       BNE   $0C3B
0C34-       A5 F7       LDA   $F7
0C36-       18          CLC
0C37-       69 04       ADC   #$04
0C39-       85 F7       STA   $F7
0C3B-       EE 0C 03    INC   $030C
0C3E-       AD 0C 03    LDA   $030C
0C41-       C9 01       CMP   #$01
0C43-       D0 07       BNE   $0C4C
0C45-       A5 F7       LDA   $F7
```

```
0C47-   85 F2       STA  $F2
0C49-   4C B2 0D    JMP  $0DB2
0C4C-   C9 02       CMP  #$02
0C4E-   D0 2B       BNE  $0C7B
0C50-   A5 F7       LDA  $F7
0C52-   18          CLC
0C53-   2A          ROL
0C54-   2A          ROL
0C55-   2A          ROL
0C56-   18          CLC
0C57-   65 F2       ADC  $F2
0C59-   85 F2       STA  $F2
0C5B-   C9 08       CMP  #$08
0C5D-   90 03       BCC  $0C62
0C5F-   4C B2 0D    JMP  $0DB2
0C62-   18          CLC
0C63-   69 08       ADC  #$08
0C65-   A0 00       LDY  #$00
0C67-   91 FC       STA  ($FC),Y
0C69-   E6 FC       INC  $FC
0C6B-   D0 02       BNE  $0C6F
0C6D-   E6 FD       INC  $FD
0C6F-   A9 18       LDA  #$18
0C71-   85 F2       STA  $F2
0C73-   A9 02       LDA  #$02
0C75-   8D 0C 03    STA  $030C
0C78-   4C B2 0D    JMP  $0DB2
0C7B-   A5 F7       LDA  $F7
0C7D-   C9 04       CMP  #$04
0C7F-   90 05       BCC  $0C86
0C81-   A5 F2       LDA  $F2
0C83-   4C 94 0C    JMP  $0C94
0C86-   A5 F7       LDA  $F7
0C88-   18          CLC
0C89-   2A          ROL
0C8A-   2A          ROL
0C8B-   2A          ROL
```

```
0C8C-      2A          ROL
0C8D-      2A          ROL
0C8E-      2A          ROL
0C8F-      18          ROL
0C90-      65 F2       ADC   $F2
0C92-      85 F2       STA   $F2
0C94-      A0 00       LDY   #$00
0C96-      91 FC       STA   ($FC),Y
0C98-      E6 FC       INC   $FC
0C9A-      D0 02       BNE   $0C9E
0C9C-      E6 FD       INC   $FD
0C9E-      A5 F7       LDA   $F7
0CA0-      F0 0C       BEQ   $0CAE
0CA2-      C9 04       CMP   #$04
0CA4-      B0 08       BCS   $0CAE
0CA6-      A9 00       LDA   #$00
0CA8-      8D 0C 03    STA   $030C
0CAB-      4C B2 0D    JMP   $0DB2
0CAE-      A9 01       LDA   #$01
0CB0-      8D 0C 03    STA   $030C
0CB3-      A5 F7       LDA   $F7
0CB5-      85 F2       STA   $F2
0CB7-      4C B2 0D    JMP   $0DB2
```

程序说明：

$0C2E－$0C40：带点矢量值加4。

$0C41－$0C4B：位元的第一部分。

$0C4C－$0C7A：位元的第二部分。

$0C86－$0CAD：位元的第三部分。

一个造型矢量对应0到7之间的一个数值，把2个或3个矢量组合成1个0到255之间的数，这个数就是造型的一个位元组。

由于一个位元组最多可存放3个矢量，因此其值等于第一个矢量加第二个矢量的8倍再加第三个矢量的64倍。

六、绘点

见程序GP 3309清单。

```
GP3309

0CBA-    A5 FE       LDA   $FE
0CBC-    38          SEC
0CBD-    E9 03       SBC   #$03
0CBF-    AA          TAX
0CC0-    A5 FF       LDA   $FF
0CC2-    38          SEC
0CC3-    E9 01       SBC   #$01
0CC5-    A0 00       LDY   #$00
0CC7-    20 57 F4    JSR   $F457
0CCA-    A5 FF       LDA   $FF
0CCC-    38          SEC
0CCD-    E9 01       SBC   #$01
0CCF-    A8          TAY
0CD0-    A2 00       LDX   #$00
0CD2-    A5 FE       LDA   $FE
0CD4-    18          CLC
0CD5-    69 03       ADC   #$03
0CD7-    20 3A F5    JSR   $F53A
0CDA-    AD BE 0C    LDA   $0CBE
0CDD-    C9 03       CMP   #$03
0CDF-    D0 07       BNE   $0CE8
0CE1-    C6 FF       DEC   $FF
0CE3-    20 E8 0C    JSR   $0CE8
0CE6-    E6 FF       INC   $FF
0CE8-    A5 FE       LDA   $FE
0CEA-    38          SEC
0CEB-    E9 03       SBC   #$03
0CED-    AA          TAX
0CEE-    A0 00       LDY   #$00
0CF0-    A5 FF       LDA   $FF
0CF2-    18          CLC
0CF3-    69 01       ADC   #$01
```

```
0CF5-    20 57 F4    JSR   $F457
0CF8-    A5 FF       LDA   $FF
0CFA-    18          CLC
0CFB-    69 01       ADC   #$01
0CFD-    A8          TAY
0CFE-    A2 00       LDX   #$00
0D00-    A5 FE       LDA   $FE
0D02-    18          CLC
0D03-    69 03       ADC   #$03
0D05-    20 3A F5    JSR   $F53A
0D08-    60          RTS
0D09-    8D BE 0C    STA   $0CBE
0D0C-    8D D6 0C    STA   $0CD6
0D0F-    8D EC 0C    STA   $0CEC
0D12-    8D 04 0D    STA   $0D04
0D15-    60          RTS
0D16-    8D C4 0C    STA   $0CC4
0D19-    8D CE 0C    STA   $0CCE
0D1C-    8D F4 0C    STA   $0CF4
0D1F-    8D FC 0C    STA   $0CFC
0D22-    20 BA 0C    JSR   $0CBA
0D25-    60          RTS
0D26-    A9 01       LDA   #$01
0D28-    20 09 0D    JSR   $0D09
0D2B-    A9 03       LDA   #$03
0D2D-    20 16 0D    JSR   $0D16
0D30-    A9 02       LDA   #$02
0D32-    20 09 0D    JSR   $0D09
0D35-    20 16 0D    JSR   $0D16
0D38-    A9 03       LDA   #$03
0D3A-    20 09 0D    JSR   $0D09
0D3D-    A9 01       LDA   #$01
0D3F-    20 16 0D    JSR   $0D16
0D42-    A5 FE       LDA   $FE
0D44-    C5 FA       CMP   $FA
0D46-    D0 06       BNE   $0D4E
0D48-    A5 FF       LDA   $FF
```

```
0D4A-      C5 FB       CMP   $FB
0D4C-      F0 17       BEQ   $0D65
0D4E-      A2 01       LDX   #$01
0D50-      20 30 F7    JSR   $F730
0D53-      A5 FF       LDA   $FF
0D55-      A6 FE       LDX   $FE
0D57-      A0 00       LDY   #$00
0D59-      20 11 F4    JSR   $F411
0D5C-      A9 00       LDA   #$00
0D5E-      A6 1A       LDX   $1A
0D60-      A4 1B       LDY   $1B
0D62-      20 5D F6    JSR   $F65D
0D65-      60          RTS
```

程序说明：

$0CBA—$0CC9：画出起始点。

$0CCA—$0CD9：连结起始点与结束点。

$0CE6—$0CF7：画出起始点。

$0CF8—$0D08：连结起始点与结束点。

$0D09—$0D41：参数设置和调用。

为了画出造型必须在表格内绘点，这个画出的点是一个实心圆，由大根线段构成：第一和第六根等长；第二和第五根等长，比第一根稍长；第三和第四根等长，比第二根稍长。

线段的画法我们在前面已作过介绍，即先定好起始点调用$F457画出，再定好结束点调用$F53A连结始点和终点形成线段。

七、造型结束

见程序GP3310清单。

GP3310

```
0D66-      A5 F6       LDA   $F6
0D68-      C9 D0       CMP   #$D0
```

```
0D6A-    D0 28       BNE   $0D94
0D6C-    AD 0C 03    LDA   $030C
0D6F-    C9 02       CMP   #$02
0D71-    D0 0C       BNE   $0D7F
0D73-    A0 00       LDY   #$00
0D75-    A5 F2       LDA   $F2
0D77-    91 FC       STA   ($FC),Y
0D79-    E6 FC       INC   $FC
0D7B-    D0 02       BNE   $0D7F
0D7D-    E6 FD       INC   $FD
0D7F-    AD 0C 03    LDA   $030C
0D82-    C9 01       CMP   #$01
0D84-    D0 0A       BNE   $0D90
0D86-    A5 F2       LDA   $F2
0D88-    18          CLC
0D89-    69 20       ADC   #$20
0D8B-    85 F2       STA   $F2
0D8D-    4C 94 0D    JMP   $0D94
0D90-    A9 04       LDA   #$04
0D92-    85 F2       STA   $F2
0D94-    A0 00       LDY   #$00
0D96-    A5 F2       LDA   $F2
0D98-    91 FC       STA   ($FC),Y
0D9A-    E6 FC       INC   $FC
0D9C-    D0 02       BNE   $0DA0
0D9E-    E6 FD       INC   $FD
0DA0-    A9 00       LDA   #$00
0DA2-    91 FC       STA   ($FC),Y
0DA4-    E6 FC       INC   $FC
0DA6-    D0 02       BNE   $0DAA
0DA8-    E6 FD       INC   $FD
0DAA-    A9 00       LDA   #$00
0DAC-    EA          NOP
0DAD-    EA          NOP
0DAE-    EA          NOP
0DAF-    4C FE 0D    JMP   $0DFE
0DB2-    A5 F5       LDA   $F5
```

```
0DB4-    C9 01       CMP   #$01
0DB6-    F0 17       BEQ   $0DCF
0DB8-    A2 01       LDX   #$01
0DBA-    20 30 F7    JSR   #F730
0DBD-    A5 F9       LDA   $F9
0DBF-    A6 F8       LDX   $F8
0DC1-    A0 00       LDY   #$00
0DC3-    20 11 F4    JSR   $F411
0DC6-    A9 00       LDA   #$00
0DC8-    A6 1A       LDX   $1A
0DCA-    A4 1B       LDY   $1B
0DCC-    20 5D F6    JSR   $F65D
0DCF-    A5 FE       LDA   $FE
0DD1-    85 F8       STA   $F8
0DD3-    A5 FF       LDA   $FF
0DD5-    85 F9       STA   $F9
0DD7-    A9 00       LDA   #$00
0DD9-    85 F5       STA   $F5
0DDB-    A2 01       LDX   #$01
0DDD-    20 30 F7    JSR   $F730
0DE0-    A5 FF       LDA   $FF
0DE2-    A6 FE       LDX   $FE
0DE4-    A0 00       LDY   #$00
0DE6-    20 11 F4    JSR   $F411
0DE9-    A9 00       LDA   #$00
0DEB-    A6 1A       LDX   $1A
0DED-    A4 1B       LDY   $1B
0DEF-    20 5D F6    JSR   $F65D
0DF2-    A5 F6       LDA   $F6
0DF4-    85 F4       STA   $F4
0DF6-    AD 0D 03    LDA   $030D
0DF9-    85 F6       STA   $F6
0DFB-    4C 41 0B    JMP   $0B41
```

程序说明：

$0D6C—$0D93：算出最后一个位元组。

$0D94—$0DB1：存入造型表。

$0DB2—$0DCE：清除光标造型。

$0DCF—$0DF1：显示光标造型。

八、加入造型表

见程序GP3311清单。

```
0DFE-    20 58 FC     JSR    $FC58
0E01-    20 2F FB     JSR    $FB2F
0E04-    A2 00        LDX    #$00
0E06-    BD C1 13     LDA    $13C1,X
0E09-    20 ED FD     JSR    $FDED
0E0C-    E8           INX
0E0D-    E0 1D        CPX    #$1D
0E0F-    D0 F5        BNE    $0E06
0E11-    20 0C FD     JSR    $FD0C
0E14-    C9 CE        CMP    #$CE
0E16-    D0 03        BNE    $0E1B
0E18-    4C 5E 0E     JMP    $0E5E
0E1B-    C9 D9        CMP    #$D9
0E1D-    D0 F2        BNE    $0E11
0E1F-    AD 0B 03     LDA    $030B
0E22-    C9 01        CMP    #$01
0E24-    EA           NOP
0E25-    EA           NOP
0E26-    AD 00 60     LDA    $6000
0E29-    CD 01 60     CMP    $6001
0E2C-    B0 03        BCS    $0E31
0E2E-    4C C6 09     JMP    $09C6
0E31-    F0 28        BEQ    $0E5B
0E33-    A9 60        LDA    #$60
0E35-    85 B9        STA    $B9
0E37-    A9 02        LDA    #$02
0E39-    85 B8        STA    $B8
0E3B-    AD 01 60     LDA    $6001
0E3E-    18           CLC
0E3F-    2A           ROL
0E40-    90 02        BCC    $0E44
```

```
0E42-   E6 B9       INC   $B9
0E44-   18          CLC
0E45-   65 B8       ADC   $B8
0E47-   85 B8       STA   $B8
0E49-   90 02       BCC   $0E4D
0E4B-   E6 B9       INC   $B9
0E4D-   A0 00       LDY   #$00
0E4F-   A5 FC       LDA   $FC
0E51-   91 B8       STA   ($B8),Y
0E53-   C8          INY
0E54-   A5 FD       LDA   $FD
0E56-   38          SEC
0E57-   E9 60       SBC   #$60
0E59-   91 B8       STA   ($B8),Y
0E5B-   4C F9 0F    JMP   $0FF9
0E5E-   CE 01 60    DEC   $6001
0E61-   4C 31 08    JMP   $0831
```

程序说明：

$0DFE—$0E13：是否存入造型表中？

$0E14—$0E1A：如不存，转$0E5E。

$0E33—$0E5D：计算下一造型的存放地址，并存入目录区。

在造型设计时如按入Q键则表示结束，返回文本态，显示：

"PUT THIS SHAPE IN THE TABLE(Y/N)?"

按入Y或N（不必回车）。如按入Y则该造型存入表中，根据当前位置指针$FC、$FD内的值算出下一个造型的起始位置，并存入造型表目录区的特定位置。

第五节 造型删除

一般的造型设计软件没有造型删除和造型内插的功能。这不仅因为删除和内插造型的过程比较复杂，还因为高级语

言的内存搬移速度太慢，难以实现对整个造型表内容的前后移动。

在本软件中我们设计了这一功能，使操作者能够任意地从造型表中删除一个造型或在表中间插入一个造型。

造型删除，见程序GP3312清单。

```
        GP3312

0E64-    20 58 FC     JSR    $FC58
0E67-    20 2F FB     JSR    $FB2F
0E6A-    A2 00
0E6C-    BD DE 13     LDA    $13DE,X
0E6F-    20 ED FD     JSR    $FDED
0E72-    E8           INX
0E73-    E0 11        CPX    #$11
0E75-    D0 F5        BNE    $0E6C
0E77-    20 6F FD     JSR    $FD6F
0E7A-    A9 00        LDA    #$00
0E7C-    85 B8        STA    $B8
0E7E-    A9 02        LDA    #$02
0E80-    85 B9        STA    $B9
0E82-    A0 00        LDY    #$00
0E84-    B1 B8        LDA    ($B8),Y
0E86-    C9 8D        CMP    #$8D
0E88-    F0 07        BEQ    $0E91
0E8A-    29 7F        AND    #$7F
0E8C-    91 B8        STA    ($B8),Y
0E8E-    C8           INY
0E8F-    D0 F3        BNE    $0E84
0E91-    20 67 DD     JSR    $DD67
0E94-    20 52 E7     JSR    $E752
0E97-    A5 51        LDA    $51
0E99-    F0 03        BEQ    $0E9E
0E9B-    4C 64 0E     JMP    $0E64
0E9E-    AD 01 60     LDA    $6001
0EA1-    C5 50        CMP    $50
```

```
0EA3-   B0 03      BCS  $0EA8
0EA5-   4C 3E 08   JMP  $083E
0EA8-   D0 03      BNE  $0EAD
0EAA-   4C 51 0F   JMP  $0F51
0EAD-   4C EC 0E   JMP  $0EEC
0EB0-   A9 60      LDA  #$60
0EB2-   85 B9      STA  $B9
0EB4-   A5 50      LDA  $50
0EB6-   18         CLC
0EB7-   2A         ROL
0EB8-   85 B8      STA  $B8
0EBA-   90 02      BCC  $0EBE
0EBC-   E6 B9      INC  $B9
0EBE-   A0 00      LDY  #$00
0EC0-   B1 B8      LDA  ($B8),Y
0EC2-   85 FE      STA  $FE
0EC4-   C8         INY
0EC5-   B1 B8      LDA  ($B8),Y
0EC7-   85 FF      STA  $FF
0EC9-   60         RTS
0ECA-   A9 60      LDA  #$60
0ECC-   85 B9      STA  $B9
0ECE-   A5 50      LDA  $50
0ED0-   18         CLC
0ED1-   2A         ROL
0ED2-   85 B8      STA  $B8
0ED4-   90 02      BCC  $0ED8
0ED6-   E6 B9      INC  $B9
0ED8-   A0 00      LDY  #$00
0EDA-   B1 B8      LDA  ($B8),Y
0EDC-   C8         INY
0EDD-   38         SEC
0EDE-   E5 F8      SBC  $F8
0EE0-   48         PHA
0EE1-   B1 B8      LDA  ($B8),Y
0EE3-   E5 F9      SBC  $F9
0EE5-   91 B8      STA  ($B8),Y
```

```
0EE7-   68          PLA
0EE8-   88          DEY
0EE9-   91 B8       STA   ($B8),Y
0EEB-   60          RTS
0EEC-   20 B0 0E    JSR   $0EB0
0EEF-   A5 FE       LDA   $FE
0EF1-   85 FA       STA   $FA
0EF3-   A5 FF       LDA   $FF
0EF5-   85 FB       STA   $FB
0EF7-   E6 50       INC   $50
0EF9-   20 B0 0E    JSR   $0EB0
0EFC-   A5 FE       LDA   $FE
0EFE-   38          SEC
0EFF-   E5 FA       SBC   $FA
0F01-   85 F8       STA   $F8
0F03-   A5 FF       LDA   $FF
0F05-   E5 FB       SBC   $FB
0F07-   85 F9       STA   $F9
0F09-   20 57 0F    JSR   $0F57
0F0C-   A5 50       LDA   $50
0F0E-   CD 01 60    CMP   $6001
0F11-   90 03       BCC   $0F16
0F13-   4C 1E 0F    JMP   $0F1E
0F16-   E6 50       INC   $50
0F18-   20 CA 0E    JSR   $0ECA
0F1B-   4C 0C 0F    JMP   $0F0C
0F1E-   A9 60       LDA   #$60
0F20-   85 FF       STA   $FF
0F22-   85 FB       STA   $FB
0F24-   EA          NOP
0F25-   EA          NOP
0F26-   EA          NOP
0F27-   EA          NOP
0F28-   A0 00       LDY   #$00
0F2A-   B1 FE       LDA   ($FE),Y
0F2C-   F0 11       BEQ   $0F3F
0F2E-   91 FA       STA   ($FA),Y
```

```
0F30-   E6 FE       INC   $FE
0F32-   D0 02       BNE   $0F36
0F34-   E6 FF       INC   $FF
0F36-   E6 FA       INC   $FA
0F38-   D0 02       BNE   $0F3C
0F3A-   E6 FB       INC   $FB
0F3C-   4C 2A 0F    JMP   $0F2A
0F3F-   C8          INY
0F40-   B1 FE       LDA   ($FE),Y
0F42-   F0 06       BEQ   $0F4A
0F44-   88          DEY
0F45-   A9 00       LDA   #$00
0F47-   4C 2E 0F    JMP   $0F2E
0F4A-   A9 00       LDA   #$00
0F4C-   91 FA       STA   ($FA),Y
0F4E-   88          DEY
0F4F-   91 FA       STA   ($FA),Y
0F51-   CE 01 60    DEC   $6001
0F54-   4C 3E 08    JMP   $083E
0F57-   A5 50       LDA   $50
0F59-   85 51       STA   $51
0F5B-   A0 02       LDY   #$02
0F5D-   B1 B8       LDA   ($B8),Y
0F5F-   A0 00       LDY   #$00
0F61-   91 B8       STA   ($B8),Y
0F63-   A0 03       LDY   #$03
0F65-   B1 B8       LDA   ($B8),Y
0F67-   A0 01       LDY   #$01
0F69-   91 B8       STA   ($B8),Y
0F6B-   E6 51       INC   $51
0F6D-   A5 51       LDA   $51
0F6F-   CD 01 60    CMP   $6001
0F72-   F0 02       BEQ   $0F76
0F74-   B0 0F       BCS   $0F85
0F76-   E6 B8       INC   $B8
0F78-   D0 02       BNE   $0F7C
0F7A-   E6 B9       INC   $B9
0F7C-   E6 B8       INC   $B8
```

```
0F7E-   D0 02      BNE  $0F82
0F80-   E6 B9      INC  $B9
0F82-   4C 5B 0F   JMP  $0F5B
0F85-   20 CA 0E   JSR  $0ECA
0F88-   60         RTS
```

程序说明：

$0E64—$0E79：输入要删除的造型序号。

$0E7A—$0E9D：转换成十六进制。

$0EB0—$0EC9：取得造型存取始址。

$0ECA—$0EEB：调整目录区有关指针。

$0F28—$0F56：目录区指针搬移。

$0F57—$0F88：造型搬移。

首先，在文本态下提问：

“NUMBER OF SHAPE?”

要求输入要删除的造型序号，并将之转换成十六进制存入单元$50。如果输入序号大于255,则必须重新输入；如果大于当前造型表内拥有的造型数，则立即返回。

造型的删除分两个部分：一是目录区指针的调整和搬移，它将被删除造型的地址指针之后直至第一个造型开始的各指针值进行调整，并全部前移两个单元，以覆盖这一造型的地址指针；二是有关造型的搬移，它将被删除造型之后直至造型表尾的所有内容前移，以覆盖这一造型。

第六节 造型内插

和造型删除类似，造型内插也需要对造型表目录区及有关造型的存放进行调整，以便插入新的造型。设计这一功能

可以使一些系统性的造型更加完整、规范，使用更加方便。

为了加快软件的执行速度，简化软件编号。我们在这一程序段片加入一个有效的技巧，这就是将造型直接加到造型表尾，而只调整目录区的结构，见程序GP 3313清单。

GP3313

```
0F89-   C9 B5       CMP   #$B5
0F8B-   D0 03       BNE   $0F90
0F8D-   4C 64 0E    JMP   $0E64
0F90-   C9 B6       CMP   #$B6
0F92-   F0 03       BEQ   $0F97
0F94-   4C 6B 10    JMP   $106B
0F97-   20 58 FC    JSR   $FC58
0F9A-   20 2F FB    JSR   $FB2F
0F9D-   A2 00       LDX   #$00
0F9F-   BD DE 13    LDA   $13DE,X
0FA2-   20 ED FD    JSR   $FDED
0FA5-   E8          INX
0FA6-   E0 11       CPX   #$11
0FA8-   D0 F5       BNE   $0F9F
0FAA-   20 6F FD    JSR   $FD6F
0FAD-   A2 00       LDX   #$00
0FAF-   BD 00 02    LDA   $0200,X
0FB2-   C9 8D       CMP   #$8D
0FB4-   F0 09       BEQ   $0FBF
0FB6-   29 7F       AND   #$7F
0FB8-   9D 00 02    STA   $0200,X
0FBB-   E8          INX
0FBC-   4C AF 0F    JMP   $0FAF
0FBF-   E0 00       CPX   #$00
0FC1-   D0 03       BNE   $0FC6
0FC3-   4C 97 0F    JMP   $0F97
0FC6-   A9 00       LDA   #$00
0FC8-   85 B8       STA   $B8
0FCA-   A9 02       LDA   #$02
```

```
0FCC-   85 B9       STA   $B9
0FCE-   20 67 DD    JSR   $DD67
0FD1-   20 52 E7    JSR   $E752
0FD4-   A5 51       LDA   $51
0FD6-   D0 BF       BNE   $0F97
0FD8-   A5 50       LDA   $50
0FDA-   CD 00 60    CMP   $6000
0FDD-   F0 0C       BEQ   $0FEB
0FDF-   90 03       BCC   $0FE4
0FE1-   4C 97 0F    JMP   $0F97
0FE4-   CD 01 60    CMP   $6001
0FE7-   F0 05       BEQ   $0FEE
0FE9-   90 03       BCC   $0FEE
0FEB-   4C BE 09    JMP   $09BE
0FEE-   8D 0F 03    STA   $030F
0FF1-   A9 E0       LDA   #$E0
0FF3-   8D 0B 03    STA   $030B
0FF6-   4C BE 09    JMP   $09BE
0FF9-   AD 0B 03    LDA   $030B
0FFC-   C9 80       CMP   #$80
0FFE-   B0 03       BCS   $1003
1000-   4C 3E 08    JMP   $083E
1003-   A9 00       LDA   #$00
1005-   8D 0B 03    STA   $030B
1008-   AD 0F 03    LDA   $030F
100B-   85 50       STA   $50
100D-   AD 01 60    LDA   $6001
1010-   18          CLC
1011-   2A          ROL
1012-   85 B8       STA   $B8
1014-   A9 60       LDA   #$60
1016-   69 00       ADC   #$00
1018-   85 B9       STA   $B9
101A-   A0 00       LDY   #$00
101C-   B1 B8       LDA   ($B8),Y
101E-   85 FE       STA   $FE
1020-   C8          INY
```

```
1021-   B1 B8       LDA   ($B8),Y
1023-   85 FF       STA   $FF
1025-   AD 01 60    LDA   $6001
1028-   85 51       STA   $51
102A-   C6 51       DEC   $51
102C-   A5 51       LDA   $51
102E-   18          CLC
102F-   2A          ROL
1030-   85 B8       STA   $B8
1032-   A9 60       LDA   #$60
1034-   69 00       ADC   #$00
1036-   85 B9       STA   $B9
1038-   A0 00       LDY   #$00
103A-   B1 B8       LDA   ($B8),Y
103C-   A0 02       LDY   #$02
103E-   91 B8       STA   ($B8),Y
1040-   A0 01       LDY   #$01
1042-   B1 B8       LDA   ($B8),Y
1044-   A0 03       LDY   #$03
1046-   91 B8       STA   ($B8),Y
1048-   A5 51       LDA   $51
104A-   C5 50       CMP   $50
104C-   F0 03       BEQ   $1051
104E-   4C 2A 10    JMP   $102A
1051-   A5 50       LDA   $50
1053-   18          CLC
1054-   2A          ROL
1055-   85 B8       STA   $B8
1057-   A9 60       LDA   #$60
1059-   69 00       ADC   #$00
105B-   85 B9       STA   $B9
105D-   A0 00       LDY   #$00
105F-   A5 FE       LDA   $FE
1061-   91 B8       STA   ($B8),Y
1063-   C8          INY
1064-   A5 FF       LDA   $FF
1066-   91 B8       STA   ($B8),Y
1068-   4C 3E 08    JMP   $083E
```

程序说明：

$0F97—$0FAC：输入内插序号。

$0FAD—$0FD7：转换成十六进制。

$0FF1—$0FF5：建立内插标志。

$1003—$1067：目录区调整。

程序GP3313首先要求用户输入内插的序号，并将之转换成十六进制存入单元$50，然后建立好内插状态的标志，转向造型建立程序。

待造型结束后，对造型表目录区内容进行调整，将内插造型的存放地址指针插入特定的位置。

第七节 图形设计

当建立好一个造型表后，通过对表中各造型的旋转、放大、移动等手段设计出各种各样的图形，这就是图形设计功能的目的。

设计好的图形可以用：

BSAVE 〈文件名〉，A$2000，L$2000↙

指令作为二进制资料形式存入磁盘，以后可随时调出使用。

一、造型的显示、放大和旋转

见程序GP3314清单。

```
GP3314

106B-    20 58 FC     JSR    $FC58
106E-    20 2F FB     JSR    $FB2F
1071-    A2 00        LDX    #$00
1073-    BD F0 13     LDA    $13F0,X
1076-    20 ED FD     JSR    $FDED
1079-    E8           INX
```

```
107A-   E0 56       CPX   #$56
107C-   D0 F5       BNE   $1073
107E-   A2 00       LDX   #$00
1080-   BD 8F 13    LDA   $138F,X
1083-   20 ED FD    JSR   $FDED
1086-   E8          INX
1087-   E0 1B       CPX   #$1B
1089-   D0 F5       BNE   $1080
108B-   20 0C FD    JSR   $FD0C
108E-   A9 01       LDA   #$01
1090-   85 E7       STA   $E7
1092-   A9 00       LDA   #$00
1094-   85 FE       STA   $FE
1096-   85 FF       STA   $FF
1098-   85 F9       STA   $F9
109A-   85 F7       STA   $F7
109C-   A9 8C       LDA   #$8C
109E-   85 F8       STA   $F8
10A0-   A9 30       LDA   #$30
10A2-   85 FA       STA   $FA
10A4-   20 D8 F3    JSR   $F3D8
10A7-   A2 03       LDX   #$03
10A9-   20 F0 F6    JSR   $F6F0
10AC-   AD 01 60    LDA   $6001
10AF-   D0 03       BNE   $10B4
10B1-   4C 20 08    JMP   $0820
10B4-   A9 00       LDA   #$00
10B6-   85 E8       STA   $E8
10B8-   A9 60       LDA   #$60
10BA-   85 E9       STA   $E9
10BC-   20 0C FD    JSR   $FD0C
10BF-   C9 C4       CMP   #$C4
10C1-   F0 03       BEQ   $10C6
10C3-   4C 14 11    JMP   $1114
10C6-   A5 F7       LDA   $F7
10C8-   F0 17       BEQ   $10E1
10CA-   A6 FE       LDX   $FE
```

```
10CC-   20 30 F7    JSR   $F730
10CF-   A9 30       LDA   #$30
10D1-   A2 8C       LDX   #$8C
10D3-   A0 00       LDY   #$00
10D5-   20 11 F4    JSR   $F411
10D8-   A6 1A       LDX   $1A
10DA-   A4 1B       LDY   $1B
10DC-   A5 FF       LDA   $FF
10DE-   20 5D F6    JSR   $F65D
10E1-   A9 01       LDA   #$01
10E3-   85 E7       STA   $E7
10E5-   A9 00       LDA   #$00
10E7-   85 FF       STA   $FF
10E9-   A9 01       LDA   #$01
10EB-   85 F7       STA   $F7
10ED-   E6 FE       INC   $FE
10EF-   A6 FE       LDX   $FE
10F1-   EC 01 60    CPX   $6001
10F4-   F0 06       BEQ   $10FC
10F6-   90 04       BCC   $10FC
10F8-   A2 01       LDX   #$01
10FA-   86 FE       STX   $FE
10FC-   20 30 F7    JSR   $F730
10FF-   A9 30       LDA   #$30
1101-   A2 8C       LDX   #$8C
1103-   A0 00       LDY   #$00
1105-   20 11 F4    JSR   $F411
1108-   A6 1A       LDX   $1A
110A-   A4 1B       LDY   $1B
110C-   A5 FF       LDA   $FF
110E-   20 01 F6    JSR   $F601
1111-   4C 69 12    JMP   $1269
1114-   C9 D3       CMP   #$D3
1116-   D0 37       BNE   $114F
1118-   A5 F7       LDA   $F7
111A-   F0 30       BEQ   $114C
111C-   A6 FE       LDX   $FE
```

```
111E-   20 30 F7    JSR   $F730
1121-   A9 30       LDA   #$30
1123-   A2 8C       LDX   #$8C
1125-   A0 00       LDY   #$00
1127-   20 11 F4    JSR   $F411
112A-   A6 1A       LDX   $1A
112C-   A4 1B       LDY   $1B
112E-   A5 FF       LDA   $FF
1130-   20 5D F6    JSR   $F65D
1133-   E6 E7       INC   $E7
1135-   A6 FE       LDX   $FE
1137-   20 30 F7    JSR   $F730
113A-   A9 30       LDA   #$30
113C-   A2 8C       LDX   #$8C
113E-   A0 00       LDY   #$00
1140-   20 11 F4    JSR   $F411
1143-   A6 1A       LDX   $1A
1145-   A4 1B       LDY   $1B
1147-   A5 FF       LDA   $FF
1149-   20 01 F6    JSR   $F601
114C    4C BC 10    JMP   $10BC
114F-   C9 D2       CMP   #$D2
1151-   D0 3C       BNE   $118F
1153-   A5 F7       LDA   $F7
1155-   F0 35       BEQ   $118C
1157-   A6 FE       LDX   $FE
1159-   20 30 F7    JSP   $F730
115C-   A9 30       LDA   #$30
115E-   A2 8C       LDX   $8C
1160-   A0 00       LDY   #$00
1162-   20 11 F4    JSR   $F411
1165-   A6 1A       LDX   $1A
1167-   A4 1B       LDY   $1B
1169-   A5 FF       LDA   $FF
116B-   20 5D F6    JSR   $F65D
116E-   A5 FF       LDA   $FF
1170-   18          CLC
```

```
1171-    69 04       ADC   #$04
1173-    85 FF       STA   $FF
1175-    A6 FE       LDX   $FE
1177-    20 30 F7    JSR   $F730
117A-    A9 30       LDA   #$30
117C-    A2 8C       LDX   #$8C
117E-    A0 00       LDY   #$00
1180-    20 11 F4    JSR   $F411
1183-    A6 1A       LDX   $1A
1185-    A4 1B       LDY   $1B
1187-    A5 FF       LDA   $FF
1189-    20 01 F6    JSR   $F601
118C-    4C BC 10    JMP   $10BC
```

程序说明：

$106B—$108D：显示提示信息。

$108E—$10A3：设置参量初值。

$10A4—$10AB：启动第二高分辨作图页，颜色代码为$03。

$10B4—$10BB：设定造型表始址为$6000。

$10BC—$10BE：接受按键。

$10BF—$1113：如按键为D，则依序显示各造型。

$1114—$114E：如按键为S，则当前造型放大。

$114F—$118E：如按键为R，则当前造型旋转。

进入图形设计功能后，首先显示按键的提示信息：

"DESIGN BY USING FOLLOWING KEYS:

D——DISPLAY, S——SCALE, R——ROT, C——CLEAR, T——ROT

Q——CLEAR, E——END, I——UP, M——DOWN, J——LEFT, K——RIGHT"

说明图形操作中11个指令键的功能和作用，然后等待按

键。

如按D键，则在屏幕上半部分显示出当前指针所指的造型，并将指针加1。如当前指针指向造型表尾，则又重新从表头开始。造型的显示位置是固定的，每显示一个造型时，先将该位置上原有的造型清除。

如按S键，则将由D键显示出来的造型进行放大，放大倍数每次加1。由于屏幕范围等的影响，当放大到一定程序时造型会发生变形。

如按R键，则将由D键显示出来的造型进行旋转，旋转角度有64个。由于造型可旋转产生图象的角度由放大倍数来决定，因此当放大倍数小于5时，某些角度不产生图象。继续按R键至一定角度时就可产生。

当用D键显示出一个造型后，可任意进行旋转和放大。

二、造型的移动、清除

见程序GP3315。

```
GP3315

118F-    C9 C3          CMP   #$C3
1191-    D0 18          BNE   $11AB
1193-    4C 8E 10       JMP   $108E
1196-    A6 FE          LDX   $FE
1198-    20 30 F7       JSR   $F730
119B-    A5 FA          LDA   $FA
119D-    A6 F8          LDX   $F8
119F-    A4 F9          LDY   $F9
11A1-    20 11 F4       JSR   $F411
11A4-    A6 1A          LDX   $1A
11A6-    A4 1B          LDY   $1B
11A8-    A5 FF          LDA   $FF
```

```
11AA-   60          RTS
11AB-   C9 C9       CMP   #$C9
11AD-   D0 20       BNE   $11CF
11AF-   A5 FA       LDA   $FA
11B1-   C9 02       CMP   #$02
11B3-   90 17       BCC   $11CC
11B5-   20 96 11    JSR   $1196
11B8-   20 5D F6    JSR   $F65D
11BB-   A5 FA       LDA   $FA
11BD-   38          SEC
11BE-   E9 02       SBC   #$02
11C0-   85 FA       STA   $FA
11C2-   20 96 11    JSR   $1196
11C5-   20 01 F6    JSR   $F601
11C8-   A9 00       LDA   #$00
11CA-   85 F7       STA   $F7
11CC-   4C BC 10    JMP   $10BC
11CF-   C9 CD       CMP   #$CD
11D1-   D0 1C       BNE   $11FF
11D3-   A5 FA       LDA   $FA
11D5-   C9 BD       CMP   #$BD
11D7-   B0 13       BCS   $11EC
11D9-   20 96 11    JSR   $1196
11DC-   20 5D F6    JSR   $F65D
11DF-   A5 FA       LDA   $FA
11E1-   18          CLC
11E2-   69 02       ADC   #$02
11E4-   85 FA       STA   $FA
11E6-   20 96 11    JSR   $1196
11E9-   20 01 F6    JSR   $F601
11EC-   4C C8 11    JMP   $11C8
11EF-   C9 CA       CMP   #$CA
11F1-   D0 20       BNE   $1213
11F3-   A5 F9       LDA   $F9
11F5-   D0 06       BNE   $11FD
11F7-   A5 F8       LDA   $F8
```

```
11F9-   C9 02       CMP   #$02
11FB-   90 EF       BCC   $11EC
11FD-   20 96 11    JSR   $1196
1200-   20 5D F6    JSR   $F65D
1203-   A5 F8       LDA   $F8
1205-   38          SEC
1206-   E9 02       SBC   #$02
1208-   85 F8       STA   $F8
120A-   A5 F9       LDA   $F9
120C-   E9 00       SBC   #$00
120E-   85 F9       STA   $F9
1210-   4C E6 11    JMP   $11E6
1213-   C9 CB       CMP   #$CB
1215-   D0 23       BNE   $123A
1217-   A5 F9       LDA   $F9
1219-   F0 09       BEQ   $1224
121B-   A5 F8       LDA   $F8
121D-   C9 15       CMP   #$15
121F-   90 03       BCC   $1224
1221-   4C C8 11    JMP   $11C8
1224-   20 96 11    JSR   $1196
1227-   20 5D F6    JSR   $F65D
122A-   A5 F8       LDA   $F8
122C-   18          CLC
122D-   69 02       ADC   #$02
122F-   85 F8       STA   $F8
1231-   A5 F9       LDA   $F9
1233-   69 00       ADC   #$00
1235-   85 F9       STA   $F9
1237-   4C E6 11    JMP   $11E6
123A-   C9 D1       CMP   #$D1
123C-   D0 09       BNE   $1247
123E-   20 96 11    JSR   $1196
1241-   20 5D F6    JSR   $F65D
1244-   4C C8 11    JMP   $11C8
1247-   C9 D4       CMP   #$D4
1249-   D0 03       BNE   $124E
```

```
124B-   4C 6E 11    JMP   $116E
124E-   C9 C5       CMP   #$C5
1250-   D0 03       BNE   $1255
1252-   4C 3E 08    JMP   $083E
1255-   20 2F FB    JSR   $FB2F
1258-   20 0C FD    JSR   $FD0C
125B-   A9 00       LDA   #$00
125D-   8D 50 C0    STA   $C050
1260-   8D 52 C0    STA   $C052
1263-   8D 55 C0    STA   $C055
1266-   8D 57 C0    STA   $C057
1269-   4C BC 10    JMP   $10BC
126C-   4C 6E 11    JMP   $116E
```

程序说明：

$118F－$1195：　如按键为C，则清除全屏幕。

$11AB－$11CE：如按键为I，则造型上移。

$11CF－$11EE：如按键为M，则造型下移。

$11EF－$1212：　如按键为J，则造型左移。

$1213－$1239：　如按键为K，则造型右移。

$123A－$1246：　如按键为Q，则清除当前造型。

$1247－$124D：　如按键为T，则造型旋转，不清除原造型。

$124E－$1254：　如按键为E，则退出。

$1255－$126E：　显示提示信息后返回。

当一个造型旋转、放大到合适的程度时，即可通过按键移动这个造型，以便构造出由若干个造型组成的用户所需要的图形。造型在移动过程中可以清除，但不能放大和旋转。

如果操作者按入一个非指令键，则转为文本状态显示前述的提示信息。按入任意一键后返回图象页，屏幕信息不

变。

第八节 资 料 表

软件的最后一部分为资料表，是程序在运行过程中所需要的参数、指令和提示信息等，由程序自动取用，见程序GP3316清单。

```
1300- C2 CC CF C1 C4 D3 C8 C1
1308- D0 C5 B0 8D C5 CE D4 C5
1310- D2 A0 D4 C8 C5 A0 CE C1
1318- CD C5 AD 00 01 00 04 00
1320- 3E 24 2D 36 04 00 00 D3
1328- C8 C1 D0 C5 A0 D4 C1 C2
1330- CC C5 A0 C9 D3 A0 C6 D5
1338- CC CC 8D 8D C2 D3 C1 D6
1340- C5 CE D5 CD C2 C5 D2 A0
1348- CF C6 A0 D3 C8 C1 D0 C5
1350- D3 A0 C9 CE A0 D4 C1 C2
1358- CC C5 BF A0 C5 CE D4 C5
1360- D2 A0 C3 CF CF D2 C4 C9
1368- CE C1 D4 C5 A0 A0 A0 CD
1370- CF D6 C5 A0 D0 CC CF D4
1378- A0 D7 C9 D4 C8 A0 C6 CF
1380- CC CC CF D7 C9 CE C7 A0
1388- CB C5 D9 D3 BA 8D 8D C9
1390- AD D5 D0 AC CD AD C4 CF
1398- D7 CE AC CA AD CC C5 C6
13A0- D4 AC CB AD D2 C9 C7 C8
13A8- D4 8D 8D D0 AD D0 CC CF
13B0- D4 AC C5 AD C5 D2 C1 D3
13B8- C5 AC D1 AD D1 D5 C9 D4
13C0- 8D D0 D5 D4 A0 D4 C8 C9
13C8- D3 A0 D3 C8 C1 D0 C5 A0
13D0- C9 CE A0 D4 C8 C5 A0 D4
13D8- C1 C2 CC C5 BF A0 CE D5
```

```
13E0- CD C2 C5 D2 A0 CF C6 A0
13E8- D3 C8 C1 D0 C5 BF A0 00
13F0- 04 05 13 09 07 0E 20 02
13F8- 19 20 15 13 09 0E 07 20
1400- 06 0F 0C 0C 0F 17 09 0E
1408- 07 20 0B 05 19 13 3A 8D
1410- 8D C4 AD C4 C9 D3 D0 CC
1418- C1 D9 AC D3 AD D3 C3 C1
1420- CC C5 AC D2 AD D2 CF D4
1428- AC C3 AD C3 CC C5 C1 D2
1430- AC D4 AD D2 CF D4 8D 8D
1438- D1 AD C3 CC C5 C1 D2 AC
1440- C5 AD C5 CE C4 AC 00 00
1448- 00 00 00 00 00 00 00 00
```

第九节 操作说明

本软件占用$800—$1445共3K多的内存空间，另外还有一幅位于第一高分辨作图页上的图形资料SHAPEO。使用时先：BLOAD〈程序名〉，待程序调入内存并出现提示符时，使用&启动。以后如程序运行出现中断而返回提示符时都可用&重新启动，但内存中已建立好的造型表不再有效（在&启动前可使用键盘指令将造型表存入磁盘，以后用功能1调入）。

任何一个造型表建立的第一步都是“造型表初置”。造型表初置确定它能容纳的造型数目（由于对十进制输入码的转换是通过直接调用ROM子程序来实现的，因此输入数值时如误输成字母或非数字字符或直接回车都会导致出错而中断运行），并对有关参数进行初值化。

下面对造型建立和图形设计中的按键指令加以介绍：

一、造型建立

·I：光点向上移动一格；

·M：光点向下移动一格；

·J：光点向左移动一格；

·K：光点向右移动一格；

·P：在光点位置上绘点；

·Q：结束造型。

二、图形设计

·I：造型向上移动，步长为2，下同；

·M：造型向下移动；

·J：造型向左移动；

·K：造型向右移动；

·D：在屏幕上部显示当前指针所指的造型，并将指针加1；

·S：将显示的造型放大，放大倍数每次加1；

·R：将显示的造型旋转，旋转角度64个；

·C：清除整个屏幕；

·T：旋转造型，但不清除原来的显示，使旋转留下轨迹；

·Q：清除当前显示的造型；

·E：返回程序清单。

至此，造型设计的机器语言程序已经介绍完了。由于程序中设计了造型删除、内插、图形设计等功能，使用比较方便，特别是图形设计功能可以解决许多较复杂的绘图问题，值得读者一试。

第四章　动画设计及图形的合并、剪辑

前面几章已经介绍过，机器语言绘图有许多高级语言所没有的优点，其中一个就是动画设计。由于机器语言执行速度很快，图形的清除、计算和再绘画过程可以在很短的时间内完成。这样，动画的效果就比较好。

第一节　简单动画设计

GP3401是一个画方形框的简单BASIC程序。

```
10   HGR2
20   FOR X = 255 TO 20 STEP  - 5
30   HCOLOR= 3: GOSUB 60
40   HCOLOR= 0: GOSUB 60
50   NEXT : END
60   HPLOT X,100 TO X,120: HPLOT X
      ,120 TO X - 20,120
70   HPLOT X - 20,120 TO X - 20,10
      0: HPLOT X - 20,100 TO X,100

80   RETURN
```

由于其执行速度较慢，方形框的移动速度也不快，且存在明显的闪动现象。把这个程序翻译成机器语言，见程序GP3402。

```
0300-  20  D8  F3  A9  FF  85  00  A5
0308-  00  C9  14  D0  01  60  A2  03
0310-  20  F0  F6  20  28  03  A2  00
0318-  20  F0  F6  20  28  03  A5  00
0320-  38  E9  05  85  00  4C  07  03
0328-  A6  00  A0  00  A9  64  20  57
```

```
0330- F4 A2 00 A5 00 A0 78 20
0338- 3A F5 A2 00 A5 00 38 E9
0340- 14 A0 78 20 3A F5 A5 00
0348- 38 E9 14 A2 00 A0 64 20
0350- 3A F5 A5 00 A2 00 A0 64
0358- 20 3A F5 60
```

执行程序GP 3402，方框的移动速度就相当快，而且非常圆滑。如果绘画的图形更复杂一些，那么这两种语言在动画效果上的差别就更加明显。

因此，使用机器语言来实现一般的动画设计可以达到令人满意的效果。不过编程时应该注意一点：绘画图形和清除图形的过程是完全一致的，不过清除图形时使用底色而已。因而可以先设置好颜色，再调用同一个子程序来绘画或清除图形。这样可以使程序更加简练。

以上我们简单介绍了直接使用机器语言进行动画设计的方法，它的原理是：先画出一个图形，再使用底色在同一位置上画这个图形（即清除）；按动画要求改变坐标值，重复以上过程。

如果图形比较复杂，要求移动速度更快，就必须使用页面轮换、造型表等技巧。

第二节　页面轮换

绘画一幅比较复杂的图形需要一定的时间，因而当清除这个图形时，其开始部分在屏幕上停留的时间最长，结束部分在屏幕上停留的时间最短。这种前后停留的时间差异使图形失去了整体的观看效果，而且使人感觉到绘图过程的缓慢，特别是破坏了图形整体的动态形象。

使用页面轮换可以较好地解决这个问题，其基本技巧是

控制显示页面和绘画页面，即先在绘画页上画好一幅图形，使用软体开关显示这一页面，同时在另一页上再画一个图形，画好以后再显示。这样，每一幅图形在屏幕上出现时都是完整的，其动态变化也很突出。

绘画页面是零页指针＄E6控制的。当＄E6的值为＄20时，为第一页面；当＄E6的值为＄40时，为第二页面；当＄E6的值为＄60时，为第三页面；依次类推。虽然我们通常只使用＄2000—＄3FFF的第一页面和＄4000—＄5FFF的第二页面，但作为绘图实际上还可以使用＄6000—＄7FFF的第三页面，＄8000—＄9FFF的第四页面和＄A000—＄BFFF的第五页面。其中第四、第五页面占据了DOS存贮区域。

当前要显示的页面由软开关控制。这样，画好的图形就能完整地显示出来而不致被清除：

AD54C0　LDA＄C054　显示第一页面

AD 55 C0　LDA＄C055　显示第二页面

GP3403是一个使用页面轮换来绘图的程序。

```
GP3403

0300-     20 58 FC     JSR    $FC58
0303-     20 E2 F3     JSR    $F3E2
0306-     20 D8 F3     JSR    $F3D8
0309-     A2 03        LDX    #$03
030B-     20 F0 F6     JSR    $F6F0
030E-     A9 03        LDA    #$03
0310-     85 00        STA    $00
0312-     A9 20        LDA    #$20
0314-     85 E6        STA    $E6
0316-     20 29 03     JSR    $0329
0319-     AD 54 C0     LDA    $C054
```

```
031C-    A9 40       LDA   #$40
031E-    85 E6       STA   $E6
0320-    20 29 03    JSR   $0329
0323-    AD 55 C0    LDA   $C055
0326-    4C 12 03    JMP   $0312
0329-    20 F2 F3    JSR   $F3F2
032C-    A9 78       LDA   #$78
032E-    38          SEC
032F-    E5 00       SBC   $00
0331-    AA          TAX
0332-    A0 00       LDY   #$00
0334-    A9 5A       LDA   #$5A
0336-    38          SEC
0337-    E5 00       SBC   $00
0339-    10 05       BPL   $0340
033B-    68          PLA
033C-    68          PLA
033D-    4C 00 03    JMP   $0300
0340-    20 57 F4    JSR   $F457
0343-    A9 5A       LDA   #$5A
0345-    38          SEC
0346-    E5 00       SBC   $00
0348-    A8          TAY
0349-    A9 78       LDA   #$78
034B-    18          CLC
034C-    65 00       ADC   $00
034E-    A2 00       LDX   #$00
0350-    20 3A F5    JSR   $F53A
0353-    A9 5A       LDA   #$5A
0355-    18          CLC
0356-    65 00       ADC   $00
0358-    A8          TAY
0359-    A9 78       LDA   #$78
035B-    18          CLC
035C-    65 00       ADC   $00
035E-    A2 00       LDX   #$00
0360-    20 3A F5    JSR   $F53A
0363-    A9 5A       LDA   #$5A
```

```
0365-    18          CLC
0366-    65 00       ADC   $00
0368-    A8          TAY
0369-    A9 78       LDA   #$78
036B-    38          SEC
036C-    E5 00       SBC   $00
036E-    A2 00       LDX   #$00
0370-    20 3A F5    JSR   $F53A
0373-    A9 5A       LDA   #$5A
0375-    38          SEC
0376-    E5 00       SBC   $00
0378-    A8          TAY
0379-    A9 78       LDA   #$78
037B-    38          SEC
037C-    E5 00       SBC   $00
037E-    A2 00       LDX   #$00
0380-    20 3A F5    JSR   $F53A
0383-    A5 00       LDA   $00
0385-    18          CLC
0386-    69 03       ADC   #$03
0388-    85 00       STA   $00
038A-    60           RTS
```

程序说明：

＄0300—＄0308：清除各个页面。

＄0309—＄030D：设置颜色为白（＄03）。

＄0312—＄0318：绘画第一页面。

＄0319—＄031B：显示第一页面。

＄031C—＄0322：绘画第二页面。

＄0323—＄0325：显示第二页面。

＄0329—＄032B：清除绘画页。

＄032C—＄0342：画出起始点。

＄0343—＄0382：画出各边。

＄0383—＄0389：放大。

程序GP 3403的屏幕效果是：在高分辨作图页上画出一个小方框，并且不断放大以至充满屏幕。它利用ROM中的画点、画线子程序在显示一个屏幕的同时，在另一个屏幕上画出一个方框，然后显示画好的页面，同时在原来的显示页面上画一个放大的方框。这样，显示和绘画总是在不同的页面上进行，两者不断互换，形成方框放大的动画状态。

程序中$0329处使用了ROM中的一个子程序，其始地址是$F3F2，功能是清除绘画页面为黑色。当$E6中的值为$20时，绘画页面为第一页，20F2F3指令清除第一页面。当$E6中的值为$40时，绘画页面是第二页，20F2F3指令清除第二页面。

第三节 造型动画

在本篇的前面几章已经介绍了造型和造型表的有关问题，并且指出了使用造型进行动画设计的优点。

实际上，由于造型是一张已经设计好的矢量表，调用造型是针对这个造型整体而言的，因此显示或清除的速度比较快，动画效果就比较好。见程序GP 3404清单。

```
GP3404

0300-   A9 44       LDA   #$44
0302-   85 E8       STA   $E8
0304-   A9 03       LDA   #$03
0306-   85 E9       STA   $E9
0308-   20 D8 F3    JSR   $F3D8
030B-   A2 03       LDX   #$03
030D-   20 F0 F6    JSR   $F6F0
0310-   A9 01       LDA   #$01
0312-   85 E7       STA   $E7
```

```
0314-   A9 AA       LDA   #$AA
0316-   85 00       STA   $00
0318-   20 2C 03    JSR   $032C
031B-   20 2C 03    JSR   $032C
031E-   C6 00       DEC   $00
0320-   A5 00       LDA   $00
0322-   C9 0A       CMP   #$0A
0324-   F0 03       BEQ   $0329
0326-   4C 18 03    JMP   $0318
0329-   4C 14 03    JMP   $0314
032C-   A2 01       LDX   #$01
032E-   20 30 F7    JSR   $F730
0331-   A2 64       LDX   #$64
0333-   A0 00       LDY   #$00
0335-   A5 00       LDA   $00
0337-   20 11 F4    JSR   $F411
033A-   A9 00       LDA   #$00
033C-   A6 1A       LDX   $1A
033E-   A4 1B       LDY   $1B
0340-   20 5D F6    JSR   $F65D
0343-   60          RTS
0344- 01 00 04 00
0348- 75 36 2E 2D 2D 0E 76 76
0350- 76 3E 0F 18 0F 18 0F 18
0358- 3F 3F 37 36 36 3F 27 24
0360- 24 3F 3F F7 1E 1E 27 0D
0368- 18 2C 20 0D 18 2C 28 2D
0370- 2D 24 6C 6C 00
```

程序说明：

$0300—$0307：设定造型表始址为$0344。

$0308—$030F：启动第二高分辨作图页，设定颜色为白。

$0310—$0313：造型放大倍数为1。

$0314—$0317：显示起始位置的纵坐标。

$0318—$031A：显示造型。

$031B—$031D：清除造型。

$031E—$0325：移动。

$032C—$0330：设定造型指针。

$0331—$0339：确定显示位置。

$033A—$0342：显示或清除造型。

GP3404是一个使用造型来进行动画设计的简单程序，执行这个程序，屏幕上就会出现一个飞机模型，这个模型不断地从下往上移动，每移动一步前先清除原造型。

程序中使用了这样一个技巧，即显示造型和清除造型是通过同一个子程序（$32C）实现的。这是因为使用了XDRAW（$F65D）指令，程序开始执行时屏幕被清除，这时XDRAW指令取底色的补色在指定位置上显示出造型图案；继续调用XDRAW，由于位置没有改变，这时的底色是刚刚用来显示造型的颜色，它的补色是图形之外的屏幕颜色，用这一颜色显示这个造型实际上就是将原来的显示清除了。因而程序的执行表现出边移动边清除的动态效果。

第四节　图形的合并

当绘制好若干个图形后，有时我们需要把其中几个合并到同一作图页上成为一幅图形。

怎样进行这种合并呢？机器语言执行速度快的特点为我们提供了极大的方便，使这种合并能够在很短的时间内完成。

现假设有名为PIC1和PIC2的两个二进制图形文件，分别是内半径为20和60的两个圆环图（是绘画好以后用BS-

AVE 〈文件名〉，A $2000，L $2000或 BSAVE 〈文件名〉,A $4000，L $2000指令存入磁盘的)。要把它们合并在一个屏幕上可使用下列步骤：

① 把第一个图形调入第一高分辨作图页：BLOAD PIC 1，A $2000↙

② 把第二个图形调入第二高分辨作图页：BLOAD PIC 2，A $4000↙

③ 运行合并程序。这样就把它们合并到第一页上。

使用软开关(POKE —16297,0：POKE —16302,0：POKE —16304，0：POKE —16300，0）即可看到合并后的图形，如图3.4 所示。

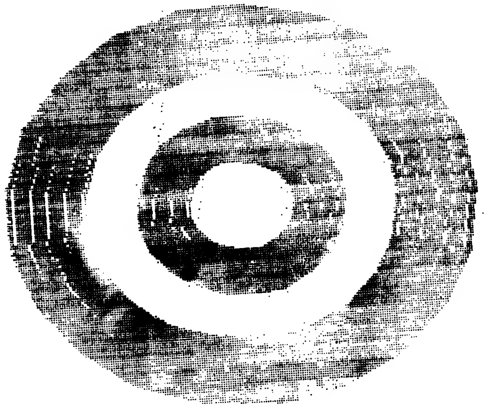


图 3.4

合并见程序GP 3405清单。

```
0300-    A9 00        LDA    #$00
0302-    85 00        STA    $00
0304-    A9 20        LDA    #$20
0306-    85 01        STA    $01
0308-    A9 00        LDA    #$00
030A-    85 02        STA    $02
030C-    A9 40        LDA    #$40
030E-    85 03        STA    $03
0310-    A0 00        LDY    #$00
0312-    B1 02        LDA    ($02),Y
0314-    F0 02        BEQ    $0318
0316-    91 00        STA    ($00),Y
0318-    C8           INY
0319-    D0 F7        BNE    $0312
031B-    E6 03        INC    $03
031D-    E6 01        INC    $01
031F-    A5 01        LDA    $01
0321-    C9 40        CMP    #$40
0323-    90 ED        BCC    $0312
0325-    60           RTS
0300-    A9 00        LDA    #$00
0302-    85 00        STA    $00
0304-    A9 40        LDA    #$40
0306-    85 01        STA    $01
0308-    A0 00        LDY    #$00
030A-    A9 FF        LDA    #$FF
030C-    F1 00        SBC    ($00),Y
030E-    91 00        STA    ($00),Y
0310-    C8           INY
0311-    D0 F7        BNE    $030A
0313-    E6 01        INC    $01
0315-    A5 01        LDA    $01
0317-    C9 60        CMP    #$60
0319-    90 EF        BCC    $030A
031B-    60           RTS
031C-    00           BRK
```

GP3405是一个简单的机器语言程序，实际上也就是一

个内存搬移程序，它将第二作图页上的非零单元搬至第二页的相应位置。

如果要合并的图形有二页以上，可以先按上述步骤合并第二和第二页，再将合并后的第二页与第三页合并，以此类推，最后把多页图形并在一个页面上。

第五节　图形的剪辑

在一个高分辨作图页上绘画就象用剪刀在一张白纸上剪出花纹一样，是一个剪、修、整的过程。基于这一原因，我们可以编写出一个相应的应用程序，且程序GP3406。

程序GP3406占用$6000—$60EF内存的空间，下面介绍一下它的主要功能和使用。

它使用一条长度可以调节的线段作为剪图的工具（实际上是一个点的造型），这条线段可以在屏幕上进行上、下、左、右的移动，并以其端点为中心不断旋转，在旋转过程中画出或清除图象。整个程序的执行速度（同时也就是线段旋转或移动的速度）可以通过按1加快或按2减慢，这样就更加方便了用户的操作。

各功能键定义如下：

① P：旋转并绘画；

② L：旋转；

③ O：旋转并清除；

④ ESC：退出运行；

⑤ I：线段上移；

⑥ M：线段下移；

⑦ J：线段左移；

⑧ K：线段右移；

⑨ H：清洗屏幕；

⑩ ←：造型缩小，即线段变短；

⑪ →：造型放大，即线段变长；

⑫ 1：加快执行；

⑬ 2：减慢执行。

在具体使用过程中，可将造型放大、缩小及移动控制的键与连续键REPT一起按下，以达到更好的效果。值得一提的是，程序中使用了延时子程序＄FCA8，这个子程序使用累加器A中的值（＄00—＄FF）作为延时参数，以控制延时的时间长短。

图3.5是用程序GP3406绘出的一幅图案。

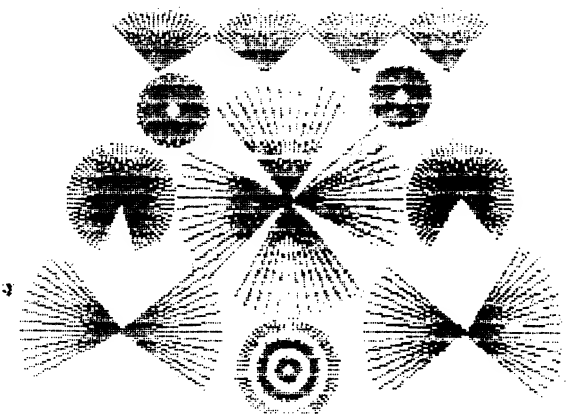


图 3.5

GP3406

```
6000-   A9 EA      LDA  #$EA
6002-   85 E8      STA  $E8
6004-   A9 60      LDA  #$60
6006-   95 E9      STA  $E9
6008-   20 D8 F3   JSR  $F3D8
600B-   A9 14      LDA  #$14
600D-   85 E7      STA  $E7
600F-   A9 8C      LDA  #$8C
6011-   85 01      STA  $01
6013-   A9 60      LDA  #$60
6015-   85 02      STA  $02
6017-   A9 01      LDA  #$01
6019-   85 00      STA  $00
601B-   AD 00 C0   LDA  $C000
601E-   A0 00      LDY  #$00
6020-   8C 10 C0   STY  $C010
6023-   C9 9B      CMP  #$9B
6025-   D0 01      BNE  $6028
6027-   60         RTS
6028-   C9 CA      CMP  #$CA
602A-   D0 02      BNE  $602E
602C-   C6 01      DEC  $01
602E-   C9 CB      CMP  #$CB
6030-   D0 02      BNE  $6034
6032-   E6 01      INC  $01
6034-   C9 C8      CMP  #$C8
6036-   D0 03      BNE  $603B
6038-   4C 08 60   JMP  $6008
603B-   C9 C9      CMP  #$C9
603D-   D0 0A      BNE  $6049
603F-   C6 02      DEC  $02
6041-   A5 02      LDA  $02
6043-   B0 04      BCS  $6049
6045-   A9 BE      LDA  #$BE
6047-   85 02      STA  $02
```

```
6049-   C9 CD      CMP  #$CD
604B-   D0 0C      BNE  $6059
604D-   E6 02      INC  $02
604F-   A5 02      LDA  $02
6051-   C9 BE      CMP  #$BE
6053-   90 04      BCC  $6059
6055-   A9 00      LDA  #$00
6057-   85 02      STA  $02
6059-   C9 88 0    CMP  #$88
605B-   D0 0C      BNE  $6069
605D-   C6 E7      DEC  $E7
605F-   A5 E7      LDA  $E7
6061-   C9 01      CMP  #$01
6063-   B0 04      BCS  $6069
6065-   A9 01      LDA  #$01
6067-   85 E7      STA  $E7
6069-   C9 95      CMP  #$95
606B-   D0 02      BNE  $606F
606D-   E6 E7      INC  $E7
606F-   C9 CC      CMP  #$CC
6071-   D0 04      BNE  $6077
6073-   A9 00      LDA  #$00
6075-   85 03      STA  $03
6077-   C9 D0      CMP  #$D0
6079-   D0 04      BNE  $607F
607B-   A9 01      LDA  #$01
607D-   85 03      STA  $03
607F-   C9 CF      CMP  #$CF
6081-   D0 04      BNE  $6087
6083-   A9 02      LDA  #$02
6085-   85 03      STA  $03
6087-   C9 B1      CMP  #$B1
6089-   D0 02      BNE  $608D
608B-   C6 04      DEC  $04
608D-   C9 B2      CMP  #$B2
608F-   D0 02      BNE  $6093
6091-   E6 04      INC  $04
```

```
6093-     A2 03          LDX   #$03
6095-     20 F0 F6       JSR   $F6F0
6098-     A5 03          LDA   $03
609A-     D0 0F          BNE   $60AB
609C-     20 D0 60       JSR   $60D0
609F-     20 5D F6       JSR   $F65D
60A2-     20 D0 60       JSR   $60D0
60A5-     20 5D F6       JST   $F65D
60A8-     4C C2 60       JMP   $60C2
60AB-     20 D0 69       JSR   $60D0
60AE-     20 01 F6       JSR   $F601
60B1-     A5 03          LDA   $03
60B3-     C9 01          CMP   #$01
60B5-     F0 0B          BEQ   $60C2
60B7-     A2 04          LDX   #$04
60B9-     20 F0 F6       JSR   $F6F0
60BC-     20 D0 60       JSR   $60D0
60BF-     20 01 F6       JSR   $F601
60C2-     E6 00          INC   $00
60C4-     A5 00          LDA   $00
60C6-     C9 40          CMP   #$40
60C8-     B0 03          BCS   $60CD
60CA-     4C 1B 60       JMP   $601B
60CD-     4C 17 60       JMP   $6017
60D0-     A5 04          LDA   $04
60D2-     20 A8 FC       JSR   $FCA8
60D5-     A2 01          LDX   #$01
60D7-     20 30 F7       JSR   $F730
60DA-     A5 02          LDA   $02
60DC-     A6 01          LDX   $01
60DE-     A0 00          LDY   #$00
60E0-     20 11 F4       JSR   $F411
60E3-     A5 00          LDA   $00
60E5-     A6 1A          LDX   $1A
60E7-     A4 1B          LDY   $1B
60E9-     60             RTS
60EA-     01 00 04 00 04 00 00 00
```

程序说明：

$6000—$600A：设定造型表始址，启动高分辨作图页。

$600B—$600E：造型放大倍数$14存入$E7中。

$600F—$6016：设定造型显示的起始位置。

$6017—$601A：设定旋转角度为$01。

$601B—$6022：读取按键，并清除。

$6023—$6027：如按入ESC，则退出。

$6028—$602D：如按入J键，则左移。

$602E—$6033：如按入K键，则右移。

$6034—$603A：如按入H键，则清除屏幕。

$603B—$6048：如按入I键，则上移。

$6049—$6058：如按入M键，则下移。

$6059—$6068：如按入左箭头，则造型缩小。

$6069—$606E：如按入右箭头，则造型放大。

$606F—$6076：如按入L键，则置$03的值为$00。

$6077—$607E：如按入P键，则置$03的值为$01。

$607F—$6086：如按入O键，则置$03的值为$02。

$6087—$608C：如按入1键，则加速。

$608D—$6092：如按入2键，则减速。

$6093—$6097：置颜色代码为$03。

$609C—$60A1：XDRAW。

$60A2—$60A7：XDRAW。

$60AB—$60B0：DRAW。

$60B7—$60C1：清除造型。

$60C2—$60CF：加大旋转角。

＄60D0—＄60D4：执行延时。

＄60D5—＄60E9：造型显示子程序。

＄60EA—＄60EF：造型表。

第六节　图形的求反

求反，这里的意思是将一幅图形（包括背景）的颜色用它的补色来替换。不同的颜色分别用它不同的补色来替换。

如果你已经在屏幕上画好一幅图形，它的颜色是白的，而背景颜色是黑的，那么，求反以后图形将是黑的，背景是白的。经过这样处理的图案在一些效果方面往往与原来不同，因而我们有时候就需要这种处理。

求反的原理非常简单，只要用＄FF去减所用图象页内的每一个单元值，把差重新存入该单元即可。以BASIC为例，对第二高分辨作图页求反只要执行语句：

〈行号〉FORI＝16384 TO 24575：POKEI，255－PEEK（I）：NEXT

其中16384到24575即第二高分辨作图页之存贮空间。但执行这样一句BASIC程序速度很慢，屏幕效果也比较差。为此，改用机器语言，它可以在极短的时间内完成求反处理，见程序GP 3407。

```
0300-     A9 00        LDA    #$00
0302-     85 00        STA    $00
0304-     A9 40        LDA    #$40
0306-     85 01        STA    $01
0308-     A0 00        LDY    #$00
030A-     A9 FF        LDA    #$FF
030C-     F1 00        SBC    ($00),Y
030E-     91 00        STA    ($00),Y
```

```
0310-   C8          INY
0311-   D0 F7       BNE    $030A
0313-   E6 01       INC    $01
0315-   A5 01       LDA    $01
0317-   C9 60       CMP    #$60
0319-   90 EF       BCC    $030A
031B-   60          RTS
031C-   00          BRK
```

如果要对第一高分辨作图页求反，应将程序中：

＄305 单元的值＄40改为＄20；

＄318 单元的值＄60改为＄40。

执行程序ＧＰ3407可通过 ＪＳＲ＄0300,呼叫。值得注意的是，当重复呼叫第二次时，图形将回到原始状态，即相当于没有求过反。

第五章 绘图程序的自动生成软件

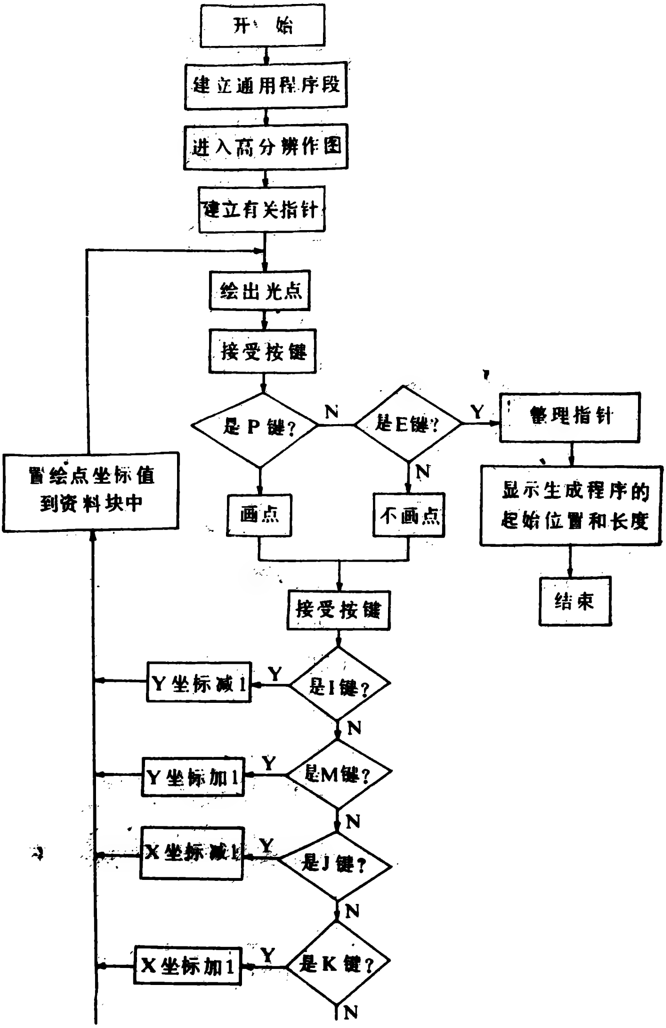
在本篇的第三部分我们介绍了一个造型的建立和管理软件。一般来说，我们可以通过调用它所建立的造型绘画出各种各样的图象。但由于各种绘图软件的具体要求不同，有时并不需要已经画好的造型，而需要绘图的程序。这种绘图程序往往通过对一系列的实验数据进行画点、画线而建立，其坐标定位繁琐，工作量大而枯燥，是程序设计者感到麻烦的事。

为此，我们用6502机器语言编制了一个使用光点移动进行作图，同时自动生成绘图程序的软件。自动生成的绘图程序也是机器语言形式的，可以存入磁盘或调入内存运行。运行结果将绘制出和生成程序时相同的图形。这样，就可以比较方便地将这一程序嵌入到应用软件中，以实现特定的绘图功能，避免了编制这种功能单一而过程复杂的程序。

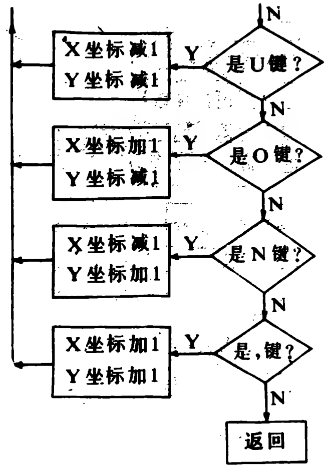
光点作用和程序自动生成软件从功能上可分为两个部分：即通过操作者的按键控制光点移动方向绘出图形和生成绘制这一图形的程序，其中后一部分是通过采集光点位置的坐标，使用通用程序段进行处理而实现的。

整个软件的运行流程（见图3.6）。

由图3.6 可见，光点在移动之前要选择一个键，决定是画点、不画点或结束。如果要画点，则点的两个坐标（用3个字节存放）要追加到通用程序段后的坐标资料块中。


开 始
建立通用程序段
进入高分辨作图
建立有关指针
绘出光点
接受按键
是P键?
N
是E键?
Y
管理指针
显示生成程序的起始位置和长度
结束
N
画点
不画点
接受按键
置绘点坐标值到资料块中
是I键?
Y
Y坐标减1
N
是M键?
Y
Y坐标加1
N
是J键?
Y
X坐标减1
N
是K键?
Y
X坐标加1
N

（a）



（b）

图 3.6

光点移动控制键共8个，根据键盘分布及习惯选择为：

1 I 键：向上　　　2 M键：向下

3 J 键：向左　　　4 K键：向右

5 U键：向左上　　6 0 键：向右上

7 N键：向左下　　8 ，键：向右下

下面我们列出清单，并对之作分节简单说明。

第一节　通用程序段和有关指针的设置

见程序GP 3501清单。

```
6000-   4C 3C 60    JMP   $603C
6003-   20 D8 F3    JSR   $F3D8
6006-   A2 03       LDX   #$03
6008-   20 F0 F6    JSR   $F6F0
```

```
600B-   A9 3C       LDA   #$3C
600D-   85 E8       STA   $E8
600F-   A9 08       LDA   #$08
6011-   85 E9       STA   $E9
6013-   A0 00       LDY   #$00
6015-   A0 00       LDY   #$00
6017-   B1 E8       LDA   ($E8),Y
6019-   AA          TAX
601A-   C8          INY
601B-   B1 E8       LDA   ($E8),Y
601D-   85 00       STA   $00
601F-   C8          INY
6020-   B1 E8       LDA   ($E8),Y
6022-   A4 00       LDY   $00
6024-   C9 FF       CMP   #$FF
6026-   D0 01       BNE   $6029
6028-   60          RTS
6029-   20 57 F4    JSR   $F457
602C-   A5 E8       LDA   $E8
602E-   18          CLC
602F-   69 03       ADC   #$03
6031-   85 E8       STA   $E8
6033-   A5 E9       LDA   $E9
6035-   69 00       ADC   #$00
6037-   85 E9       STA   $E9
6039-   4C 15 08    JMP   $0815
603C-   A0 00       LDY   #$00
603E-   B9 03 60    LDA   $6003,Y
6041-   99 03 08    STA   $0803,Y
6044-   C8          INY
6045-   C0 39       CPY   #$39
6047-   D0 F5       BNE   $603E
6049-   20 D8 F3    JSR   $F3D8
604C-   A9 3C       LDA   #$3C
604E-   85 E8       STA   $E8
6050-   A9 08       LDA   #$08
6052-   85 E9       STA   $E9
```

```
6054-    A9 90        LDA   #$90
6056-    85 EA        STA   $EA
6058-    A9 00        LDA   #$00
605A-    85 EB        STA   $EB
605C-    A9 60        LDA   #$60
605E-    85 EC        STA   $EC
```

从$6003到$603B为通用程序块。由于绘图过程中建立了坐标序列资料，因此通用程序块通过设置和调整指针就很容易取得这些坐标，并将其画出来。当遇到结束标志时则退出运行。这样，一幅完整的图象就通过有限个点表示出来。

第二节 光点控制

见程序GP 3502清单。

```
6060-    A2 03        LDX   #$03
6062-    20 F0 F6     JSR   $F6F0
6065-    A9 FF        LDA   #$FF
6067-    85 00        STA   $00
6069-    A5 EC        LDA   $EC
606B-    A4 EB        LDY   $EB
606D-    A6 EA        LDX   $EA
606F-    20 57 F4     JSR   $F457
6072-    20 35 FD     JSR   $FD35
6075-    C9 D0        CMP   #$D0
6077-    D0 04        BNE   $607D
6079-    A9 00        LDA   #$00
607B-    F0 03        BEQ   $6080
607D-    4C 02 62     JMP   $6202
6080-    85 00        STA   $00
6082-    20 35 FD     JSR   $FD35
6085-    C9 C9        CMP   #$C9
6087-    D0 16        BNE   $609F
6089-    A5 EC        LDA   $EC
608B-    F0 D3        BEQ   $6060
```

```
608D-   A5 00      LDA  $00
608F-   F0 06      BEQ  $6097
6091-   20 D5 61   JSR  $61D5
6094-   4C 9A 60   JMP  $609A
6097-   20 E4 61   JSR  $61E4
609A-   C6 EC      DEC  $EC
609C-   4C 60 60   JMP  $6060
609F-   C9 CD      CMP  #$CD
60A1-   D0 18      BNE  $60BB
60A3-   A5 EC      LDA  $EC
60A5-   C9 BE      CMP  #$BE
60A7-   B0 BA      BCS  $6063
60A9-   A5 00      LDA  $00
60AB-   F0 06      BEQ  $60B3
60AD-   20 D5 61   JSR  $61D5
60B0-   4C B6 60   JMP  $60B6
60B3-   20 E4 61   JSR  $61E4
60B6-   E6 EC      INC  $EC
60B8-   4C 60 60   JMP  $6060
60BB-   C9 CA      CMP  #$CA
60BD-   D0 25      BNE  $60E4
60BF-   A5 EA      LDA  $EA
60C1-   D0 04      BNE  $60C7
60C3-   A5 EB      LDA  $EB
60C5-   F0 99      BEQ  $6060
60C7-   A5 00      LDA  $00
60C9-   F0 06      BEQ  $60D1
60CB-   20 D5 61   JSR  $61D5
60CE-   4C D4 60   JMP  $60D4
60D1-   20 E4 61   JSR  $61E4
60D4-   A5 EA      LDA  $EA
60D6-   38         SEC
60D7-   E9 01      SBC  #$01
60D9-   85 EA      STA  $EA
60DB-   A5 EB      LDA  $EB
60DD-   E9 00      SBC  #$00
60DF-   85 EB      STA  $EB
60E1-   4C 60 60   JMP  $6060
```

```
60E4-    C9 CB        CMP   #$CB
60E6-    D0 25        BNE   $610D
60E8-    A5 EB        LDA   $EB
60EA-    C9 00        CMP   #$00
60EC-    F0 09        BEQ   $60F7
60EE-    A5 EA        LDA   $EA
60F0-    C9 18        CMP   #$18
60F2-    90 03        BCC   $60F7
60F4-    4C 60 60     JMP   $6060
60F7-    A5 00        LDA   $00
60F9-    F0 06        BEQ   $6101
60FB-    20 D5 61     JSR   $61D5
60FE-    4C 04 61     JMP   $6104
6101-    20 E4 61     JSR   $61E4
6104-    E6 EA        INC   $EA
6106-    D0 02        BNE   $610A
6108-    E6 EB        INC   $EB
610A-    4C 60 60     JMP   $6060
610D-    C9 D5        CMP   #$D5
610F-    D0 2E        BNE   $613F
6111-    A5 EC        LDA   $EC
6113-    F0 08        BEQ   $611D
6115-    A5 EB        LDA   $EB
6117-    D0 07        BNE   $6120
6119-    A5 EA        LDA   $EA
611B-    D0 03        BNE   $6120
611D-    4C 60 60     JMP   $6060
6120-    A5 00        LDA   $00
6122-    F0 06        BEQ   $612A
6124-    20 D5 61     JSR   $61D5
6127-    4C 2D 61     JMP   $612D
612A-    20 E4 61     JSR   $61E4
612D-    C6 EC        DEC   $EC
612F-    A5 EA        LDA   $EA
6131-    38           SEC
6132-    E9 01        SBC   #$01
6134-    85 EA        STA   $EA
```

```
6136-   A5 EB        LDA  $EB
6138-   E9 00        SBC  #$00
613A-   85 EB        STA  $EB
613C-   4C 60 60     JMP  $6060
613F-   C9 CF        CMP  #$CF
6141-   D0 29        BNE  $616C
6143-   A5 EC        LDA  $EC
6145-   F0 0A        BEQ  $6151
6147-   A5 EB        LDA  $EB
6149-   D0 09        BNE  $6154
614B-   A5 EA        LDA  $EA
614D-   C9 18        CMP  #$18
614F-   90 03        BCC  $6154
6151-   4C 60 60     JMP  $6060
6154-   A5 00        LDA  $00
6156-   F0 06        BEQ  $615E
6158-   20 D5 61     JSR  $61D5
615B-   4C 61 61     JMP  $6161
615E-   20 E4 61     JSR  $61E4
6161-   C6 EC        DEC  $EC
6163-   E6 EA        INC  $EA
6165-   D0 02        BNE  $6169
6167-   E6 EB        INC  $EB
6169-   4C 60 60     JMP  $6060
616C-   C9 CE        CMP  #$CE
616E-   D0 30        BNE  $61A0
6170-   A5 EC        LDA  $EC
6172-   C9 BE        CMP  #$BE
6174-   B0 08        BCS  $617E
6176-   A5 EA        LDA  $EA
6178-   D0 07        BNE  $6181
617A-   A5 EB        LDA  $EB
617C-   D0 03        BNE  $6181
617E-   4C 60 60     JMP  $6060
6181-   A5 00        LDA  $00
6183-   F0 06        BEQ  $618B
6185-   20 D5 61     JSR  $61D5
```

```
6188-   4C 8E 61    JMP   $618E
618B-   20 E4 61    JSR   $61E4
618E-   E6 EC       INC   $EC
6190-   A5 EA       LDA   $EA
6192-   38          SEC
6193-   E9 01       SBC   #$01
6195-   85 EA       STA   $EA
6197-   A5 EB       LDA   $EB
6199-   E9 00       SBC   #$00
619B-   85 EB       STA   $EB
619D-   4C 60 60    JMP   $6060
61A0-   C9 AC       CMP   #$AC
61A2-   F0 06       BEQ   $61AA
61A4-   20 3A FF    JSR   $FF3A
61A7-   4C 60 60    JMP   $6060
61AA-   A5 EC       LDA   $EC
61AC-   C9 BE       CMP   #$BE
61AE-   B0 0A       BCS   $61BA
61B0-   A5 EB       LDA   $EB
61B2-   F0 09       BEQ   $61BD
61B4-   A5 EA       LDA   $EA
61B6-   C9 18       CMP   #$18
61B8-   90 03       BCC   $61BD
61BA-   4C 60 60    JMP   $6060
61BD-   A5 00       LDA   $00
61BF-   F0 06       BEQ   $61C7
61C1-   20 D5 61    JSR   $61D5
61C4-   4C CA 61    JMP   $61CA
61C7-   20 E4 61    JSR   $61E4
61CA-   E6 EC       INC   $EC
61CC-   E6 EA       INC   $EA
61CE-   D0 02       BNE   $61D2
61D0-   E6 EB       INC   $EB
61D2-   4C 60 60    JMP   $6060
```

程序 GP 3502是软件的主要部分，它的主要功能是接受、判断按键操作，并根据不同的按键移动光点，改变光点位置

指针，采集光点坐标数据。

其中调用的两个子程序＄61D5和＄61E4分别是清除原光点位置及采集坐标数据的子程序。如果按光点移动键之前没有按P键，表明不画点，调用＄61D5子程序清除再移动光点。否则不清除，且调用＄61E4子程序取得坐标数据，再移动光点。

第三节　几个子程序

见程序GP3503清单。

```
61D5-    A2 00       LDX   #$00
61D7-    20 F0 F6    JSR   $F6F0
61DA-    A5 EC       LDA   $EC
61DC-    A4 EB       LDY   $EB
61DE-    A6 EA       LDX   $EA
61E0-    20 57 F4    JSR   $F457
61E3-    60          RTS
61E4-    A0 00       LDY   #$00
61E6-    A5 EA       LDA   $EA
61E8-    91 E8       STA   ($E8),Y
61EA-    C8          INY
61EB-    A5 EB       LDA   $EB
61ED-    91 E8       STA   ($E8),Y
61EF-    C8          INY
61F0-    A5 EC       LDA   $EC
61F2-    91 E8       STA   ($E8),Y
61F4-    A5 E8       LDA   $E8
61F6-    18          CLC
61F7-    69 03       ADC   #$03
61F9-    85 E8       STA   $E8
61FB-    A5 E9       LDA   $E9
61FD-    69 00       ADC   #$00
61FF-    85 E9       STA   $E9
6201-    60          RTS
```

```
6202-    C9 C5          CMP   #$C5
6204-    F0 05          BEQ   $620B
6206-    A9 FF          LDA   #$FF
6208-    4C 80 60       JMP   $6080
620B-    A9 FF          LDA   #$FF
620D-    A0 00          LDY   #$00
620F-    91 E8          STA   ($E8),Y
6211-    C8             INY
6212-    91 E8          STA   ($E8),Y
6214-    C8             INY
6215-    91 E8          STA   ($E8),Y
6217-    A5 E8          LDA   $E8
6219-    18             CLC
621A-    69 03          ADC   #$03
621C-    85 E8          STA   $E8
621E-    A5 E9          LDA   $E9
6220-    69 00          ADC   #$00
6222-    85 E9          STA   $E9
6224-    20 E2 F3       JSR   $F3E2
6227-    A9 16          LDA   #$16
6229-    85 25          STA   $25
622B-    20 8E FD       JSR   $FD8E
622E-    A2 00          LDX   #$00
6230-    BD 46 62       LDA   $6246,X
6233-    20 ED FD       JSR   $FDED
6236-    E8             INX
6237-    E0 13          CPX   #$13
6239-    D0 F5          BNE   $6230
623B-    A5 E9          LDA   $E9
623D-    20 DA FD       JSR   $FDDA
6240-    A5 E8          LDA   $E8
6242-    20 DA FD       JSR   $FDDA
6245-    60             RTS
6246- C2 C5
6248- C7 C9 CE BA A4 B8 B0 B3
6250- BD C5 CE C4 A0 A0 A0 BA
6258- A4 00
```

第三段结束程序是在决定绘点或不绘点时按E键转来的。它的作用有两个，即设置坐标数据序列的结束标志＄FF和显示生成程序的起始位置、结束位置：

BEGIN：＄803

END：〈结束地址〉

因此，生成程序的长度为L＝〈结束地址〉－＄803＋1，使用指令：BSAVE〈文件名〉，A＄803，L＄〈长度〉

或

BRUN〈文件名〉即可存入磁盘或调入内存运行。

至此，这一软件就介绍完了。不难发现软件有两个不够完善的地方。其一是作为程序本身不能灵活使用内存空间，而固定生成程序的起始地址为＄803；其二是一旦画好点后不能清除，这就要求按键无误，有时不易做到。

对于第一个问题，处理比较简单，只需在移动通用程序前要求用户输入存放始址，再将之移入指定地址，并根据输入地址计算出通用程序尾（即坐标数据存放的起始地址），放入地址指针＄E8、＄E9中即可。第二个问题稍复杂一些，但允许修改最后有限个绘图点却容易做到。例如可以用←键进行修改，每按一次左箭头←键就取出最后一组坐标值（3个字节）用底色绘画，同时使＄E8、＄E9指针下移3个字节。

有兴趣的读者可以就以上问题对软件进行进一步完善，在此不再赘述。

第六章　绘图与声音

前面几章主要介绍了机器语言绘图的原理、方法和技巧，并比较详细地剖析了几个实用的绘图软件。我们知道，绘图较多地使用在机械、建筑、数学和游戏程序设计中，如何在绘图过程中加入声音效果，使操作过程“有声有色”是绘图技术中值得探讨的问题。尤其是教学和游戏一类的程序，没有声音的配合就会变得枯燥无味，从而失去吸引力和应有的作用。

本章就绘图与声音方面的问题作一些初步介绍。

第一节　发声的基本原理

在计算机的主机内部有一个能够发出声音的扬声器，这个扬声器可以通过内存中特定的地址单元来改变状态，当扬声器由一种状态变为另一种状态时，就发出一声短音。这种短音很轻，一般不注意不易听见。但连续改变扬声器的状态，并设定每次发音的强度和时间时，就能组合出各种各样的模拟声音。

在 APPLE 内部，控制扬声器发音的特定地址单元是 49200（$C030）。不管是取出这个单元的数还是把一个数存入这一单元，只要触及到这一单元，扬声器就会发音。因此，在监控状态下直接输入：

C030↙　和执行指令：

8D 30 C0 STA $C030

或

AD 30 C0 LDA $C030 具有同样的效果。

顺便说明一下，使用BASIC指令和语句也能使扬声器发音，这种语句即：

PRINT PEEK(49200)

X=PEEK(49200)

或

POKE 49200,A

但由于BASIC指令执行时要经过复杂的解释过程，速度较慢，因而发音频率受到限制，难以满足实际需要。使用机器语言就可以大大提高运行速度，从而加快扬声器的振动。

第二节 机器语言发音子程序

不同的发音频率和发音时间构成了各种各样的声音效果，因此，如果能够控制发音频率的高低和发音时间的长短就能模拟出各种声音。这种控制一般是通过机器语言发音程序来实现的，其清单见GP3601清单。

```
0300-  AD 30 C0   LDA  $C030
0303-  88         DEY
0304-  D0 04      BNE  $030A
0306-  C6 01      DEC  $01
0308-  F0 08      BEQ  $0312
030A-  CA         DEX
030B-  D0 F6      BNE  $0303
```

```
030D— A6    00          LDX  $00
030F— 4C    00  03      JMP  $0300
0312—  60               RTS
```

其中零页单元$00、$01中分别存放着发音周期和发音时间值。

程序GP3601在一些参考书上有详细的介绍，在此不再探究，而着重于在绘图过程中如何使用这一子程序发出恰当的声音。

第三节 绘图与声音

绘图与声音的配合是教学、游戏一类程序中必不可少的内容，它能够增强程序的趣味性、知识性，使人同时得到视觉和听觉上的享受。根据程序在某一运行过程中的具体情况，可以模拟唱歌、说话、爆炸、碰撞等各种声音。

下面我们举几个简单的程序例子来说明。假设机器语言发音子程序已放入$300开始的内存单元中。

一、射击模拟

这一程序利用$606B—$6087的地址空间存放造型表，表中设置有手枪的造型。程序则存放在$6000—$606A的内存中，它首先进入高分辨作图状态，并在屏幕位置横坐标为$00，纵坐标为$80的地方显示出手枪图形。从$6028开始程序通过发音和子弹射出过程的交替执行，实现了较好的模拟效果。

见程序GP3602清单。

```
6000-    20 D8 F3      JSR    $F3D8
6003-    A2 03         LDX    #$03
6005-    20 F0 F6      JSR    $F6F0
```

```
6008-   A9 6B       LDA   #$6B
600A-   85 E8       STA   $E8
600C-   A9 60       LDA   #$60
600E-   85 E9       STA   $E9
6010-   A2 01       LDX   #$01
6012-   20 30 F7    JSR   $F730
6015-   A9 80       LDA   #$80
6017-   A2 00       LDX   #$00
6019-   A0 00       LDY   #$00
601B-   20 11 F4    JSR   $F411
601E-   A9 00       LDA   #$00
6020-   A6 1A       LDX   $1A
6022-   A4 1B       LDY   $1B
6024-   20 01 F6    JSR   $F601
6027-   EA          NOP
6028-   A9 01       LDA   #$01
602A-   85 00       STA   $00
602C-   85 01       STA   $01
602E-   20 00 03    JSR   $0300
6031-   E6 01       INC   $01
6033-   E6 00       INC   $00
6035-   D0 F7       BNE   $602E
6037-   A9 10       LDA   #$10
6039-   85 02       STA   $02
603B-   A2 03       LDX   #$03
603D-   20 F0 F6    JSR   $F6F0
6040-   A9 80       LDA   #$80
6042-   A0 00       LDY   #$00
6044-   A6 02       LDX   $02
6046-   20 57 F4    JSR   $F457
6049-   A9 30       LDA   #$30
604B-   20 A8 FC    JSR   $FCA8
604E-   A2 00       LDX   #$00
6050-   20 F0 F6    JSR   $F6F0
6053-   A9 80       LDA   #$80
6055-   A0 00       LDY   #$00
```

```
6057-    A6 02          LDX   $02
6059-    20 57 F4       JSR   $F457
605C-    E6 02          INC   $02
605E-    F0 08          BEQ   $6068
6060-    EA             NOP
6061-    E6 00          INC   $00
6063-    E6 01          INC   $01
6065-    4C 3B 60       JMP   $603B
6068-    4C 28 60       JMP   $6028
606B- 01 01 04 00 2D
6070- 2D 2D 2D 2D 2D 35 3F 3F
6078- 3F 3F 3F 3F 37 36 36 25
6080- 24 6C 49 5E 1E 17 3F F8
6088- 03 00 00
```

二、跳伞模拟

跳伞模拟程序的结构和上一个程序大致相同。其运行过程是一架降落伞从屏幕上端下落，直至屏幕底部，降落过程伴有有趣的声音，见程序G P 3603清单。

```
6000-    20 D8 F3       JSR   $F3D8
6003-    A2 03          LDX   #$03
6005-    20 F0 F6       JSR   $F6F0
6008-    A9 50          LDA   #$50
600A-    85 E8          STA   $E8
600C-    A9 60          LDA   #$60
600E-    85 E9          STA   $E9
6010-    A9 10          LDA   #$10
6012-    85 02          STA   $02
6014-    A2 01          LDX   #$01
6016-    20 30 F7       JSR   $F730
6019-    A5 02          LDA   $02
601B-    A2 80          LDX   #$80
601D-    A0 00          LDY   #$00
601F-    20 11 F4       JSR   $F411
```

```
6022-   A6 1A       LDX   $1A
6024-   A4 1B       LDY   $1B
6026-   A9 00       LDA   #$00
6028-   20 5D F6    JSR   $F65D
602B-   A2 01       LDX   #$01
602D-   20 30 F7    JSR   $F730
6030-   A5 02       LDA   $02
6032-   A2 80       LDX   #$80
6034-   A0 00       LDY   #$00
6036-   20 11 F4    JSR   $F411
6039-   A6 1A       LDX   $1A
603B-   A4 1B       LDY   $1B
603D-   A9 00       LDA   #$00
603F-   20 5D F6    JSR   $F65D
6042-   E6 02       INC   $02
6044-   E6 00       INC   $00
6046-   A9 10       LDA   #$10
6048-   85 01       STA   $01
604A-   20 00 03    JSR   $0300
604D-   4C 14 60    JMP   $6014

6050-  01 01 04 00 1A 0D 18 2D
6058-  28 0D 76 22 08 75 75 DF
6060-  1E 1E 1E 0F 18 0F 18 0F
6068-  18 4D 08 18 36 36 D7 0E
6070-  6E 2C B0 3A 3F 37 2D 2D
6078-  3E 3F 6F 13 3E 4D 39 6C
6080-  09 00 2D 4D 49 49 00 DB
```

不难发现，$6014—$602A和$602B—$6041是两个完全相同的程序段，也完全可以用两次呼叫这样一个子程序来代替，但其产生的效果是不同的。第一个程序段是显示跳伞造型，第二个程序段则是清除跳伞造型，因为其中用了$F65D的造型调用子程序，而不是$F601。

第七章　绘图程序的必备功能

作为本篇的一个结束，我们再谈一谈比较完善的绘图程序所应具备的一些基本功能。这种基本功能指的是除了完成计算、绘画之外的能增强程序实用性、透明性、可扩充性和方便用户操作等方面的功能。

这个问题显然没有统一的标准，但从程序设计本身以及操作者使用两个方面来考虑，不外乎以下几种基本功能。

第一节　程序结构清晰

主程序、子程序及资料段规格存放，使程序易读、易修改、易扩充。按照一般的习惯，子程序存放在一个独立的内存段中，不与其它部分混在一起；资料段则通常置于整个程序的末尾。为了节省内存空间，简化主程序的运行流程，有时表面上看来不能统一的两个程序段也可通过传送参量统一起来。

例1，见程序GP3701。

```
0300-   A2   00        LDX  #$00
0302-   20   20   03   JSR   $0320
0305-   20   35   FD   JSR   $FD35
0308-   A2   3F
030A-   20   20   03   JSR   $0320
  ⋮          ⋮
```

```
0320—  BD  30  03   LDA   $0330,X
0323—  F0  06       BEQ   $032B
0325—  20  ED  FD   JSR   $FDED
0328—  E8           INX
0329—  D0  F5       BNE   $0320
0328—  60           RTS
  ⋮         ⋮         ⋮
0330—  D0  D2  C5  D3  D3  A0  C1  CE
0338—  D9  A0  CB  C5  D9  8D  00  D7
0340—  C1  C9  D4  A0  C1  A0  CD  CF
0348—  CD  C5  CE  D4  8D  00
```

程序GP 3701的目的是先显示“PRESS ANY KEY”信息，并等待按入任一键，再显示“WAIT A MO-MENT”。显示的两行信息并不相同，本来需要两段相似但不相同的程序来完成。这里则采用了给同一子程序传送不同的资料定位指针（暂存器X）来调用同一子程序$0320。显然，这样做节省了内存，且使主程序变得简洁明了。

第二节　程序设计灵活

有暂停或退出绘图的功能。这就要求在循环圈内有键盘扫描和判别程序。例2，在一个循环圈内嵌入这样一段程序，见GP 3702。

```
0300—  AD  10  C0   LDA   $C010  ⎫
0303—  C9  80       CMP   #$80   ⎬ 没有按键，继续执行
0305—  90  13       BCC   $031A  ⎭
```

```
0307 — AD 00 C0   LDA  $C000    按下ESC键退出
030A — C9 9B      CMP  #$9B
030C — D0 01      BNE  $030F
030E — 60         RTS
030F — C9 8D      CMP  #$8D     再按回车键，继续
0311 — F0 07      BEQ  $031A
0313 — A9 00      LDA  #$00     清除
0315 — 8D 10 C0   STA  $C010
0318 — F0 E6      BEQ  $0300    按其它键，暂停
031A — ...
```

则程序每次运行到这里时对键盘进行扫描 如发现有按键，则判断

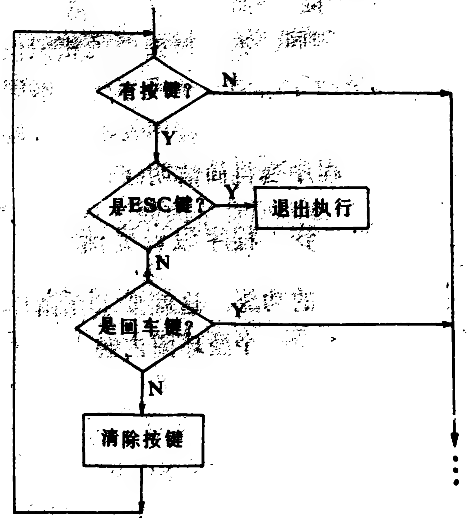


图 3.7

· 是ESC键退出运行；

· 是其它非回车键，则转＄0300继续扫描，其效果即是暂停执行；

· 在暂停时如按入回车键则继续执行。

这样，就使用户的操作具有一定的灵活性，见图3.7。

第三节　图象配有说明

做到图文并茂，使人一目了然。这种文字说明可以借助图象文本混合页面的下端显示，也可以使用造型汉字或字符在图面上显示。

第四节　提示注意事项

需要提醒用户特别注意的文字说明或图形部分要用不同颜色或交替闪烁作为标志。另外，由于机器语言执行速度很快，必要时应加入延时子程序，避免图象闪烁太快而无法看清。这种延时可以调用＄FCA8的系统程序实现：

```
××××- A9 40        LDA  #$40
××××- 20  A8  FC   JSR  $FCA8
  ⋮         ⋮
```

其中累加器A的值决定延时时间，A值为＄01时时间最短。

结　束　语

随着微型计算机的广泛应用，计算机绘图已成为一门重要的设计技术，而6502机器语言又是一种绘图功能较强的程序语言，它具有快速、灵活、方便等优点，被普遍应用在游戏程序、辅助教学、电路设计、机械、建筑设计等各个方面。

基于上述原因，本篇从6502机器语言的绘画原理出发，较为详细地介绍了它的绘画功能及相应的程序设计方法，并通过几个实用绘画软件由浅入深剖析了6502的绘图技术及多种绘图技巧。特别是造型管理等几个软件，设计较为精炼，技巧性、趣味性较强，又具有较为丰富的绘图功能，读者可以从中看到6502绘画的概貌。

应该说明的是，绘图技术已经涉及到各行各业以及各种图形设计，而本篇所介绍的仅仅是6502绘图的一般方法和技巧。因此，在解决一个实际问题时，往往需要针对具体情况融进这些方法和技巧，以达到预期的目的。

另外，为了使绘图软件具有更加清晰的人机界面，也为了软件本身的易于扩充和维护，让BASIC语言和机器语言结合起来使用，会收到更好的效果。

科技新书目：224-212　　定价：9.10元

ISBN 7-5029-0425-5/TP·0022

中华学习机图形管理

气象出版社